UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

Rafael Weingärtner

CONTROLE DE DISSEMINAÇÃO DE DADOS SENSÍVEIS EM AMBIENTES FEDERADOS

Florianópolis

2014

Rafael Weingärtner

CONTROLE DE DISSEMINAÇÃO DE DADOS SENSÍVEIS EM AMBIENTES FEDERADOS

Dissertação submetida à Pós-Graduação em Ciência da Computação para a obtenção do Grau de Mestre em Ciência da Computação.
Orientadora: Prof.ª Dra. Carla Merkle Westphall

Florianópolis

2014

RESUMO

Computação em nuvem é amplamente utilizada para fornecer serviços sob demanda, frente aos seus benefícios, como redução de custos, flexibilidade e agilidade no provisionamento de recursos. No entanto, ainda existem organizações e usuários que não estão confortáveis em enviar seus dados sensíveis para a nuvem, em especial dados de identificação, também conhecidos como *PII (personally identifiable information)*. Estes dados podem ser utilizados para identificar um usuário em ambientes externos aos sistemas em que são utilizados. Além disso, existem casos de vazamentos de dados que resultaram em roubos de identidade, sendo consequência de agentes internos (administradores de sistema maliciosos) ou externos. Este trabalho apresenta uma abordagem para tratar as questões que envolvem privacidade em torno dos PIIs armazenados em provedores de identidade (IdP). O modelo proposto busca reduzir os riscos envolvidos no momento da disseminação dos dados dos usuários ao mesmo tempo em que provê um controle mais apurado dos PIIs armazenados em IdPs. Deste modo, dificulta-se o vazamento de PIIs e apropriação indevida de dados de identificação dos usuários.

Palavras-chave: Computação em nuvem. Federação. Gerenciamento de Identidades. Privacidade.

ABSTRACT

Cloud computing is widely used to provide on demand services, mainly, because of its benefits such as reduced costs, structure flexibility and agility on resource provisioning. However, there are still companies and users that are not comfortable in sending their sensitive data to the cloud, especially the personally identifiable information (PII) that could be used to identify someone outside of the system in which they are used. Moreover, there have been cases of data leaks which resulted in identity thefts that were either consequence of internal agents (malicious system administrators) or external ones. We present a model that addresses the privacy issues within the PII that is stored in identity providers. In one hand, our proposal lowers risks when disseminating PII data and increases awareness of the process. In other hand, it gives control of PII data that is stored in identity providers directly to users' hands.

Keywords: Cloud Computing. Identity management. Federation. Privacy

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

SUMÁRIO

1 INTRODUÇÃO

Computação em nuvem está sendo amplamente adotada para fornecer serviços (HALL, 2012; FORUM, 2013). A migração para a nuvem acontece como consequência de suas características como redução de custos, elasticidade a agilidade na adição e subtração de novos recursos. No entanto, ainda existem organizações e usuários que não estão confortáveis em enviar seus dados sensíveis para a nuvem (SRINIVASAMURTHY; LIU, 2010). Sobre o assunto de computação em nuvem acontecem debates não somente sobre o aspecto da tecnologia em si, mas, sobre as questões comerciais e de governança que se relacionam com a segurança de dados e privacidade (FORUM, 2013).

A segurança e privacidade na nuvem são alvos de preocupações para usuários e organizações. Existem casos de violações de privacidade dos usuários e apropriação indevida de dados (HELFT, 2011; KOCIENIEWSKI, 2013; SANG-HUN, 2014). Han-zhang e Liu-sheng (2010) realizam uma discussão sobre os usuários de serviços em nuvem que não possuem controle sobre a infraestrutura na qual o recurso acessado está hospedado e da possibilidade de existirem administradores de sistemas curiosos/maliciosos, que possuem habilidades técnicas e permissões para se apropriar indevidamente de dados dos usuários. Deste modo, os provedores deveriam voltar-se a proteger dados sensíveis ao invés de somente reforçar as barreiras de segurança contra ameaças externas, uma vez que a maior ameaça pode ser interna.

Conforme apresentado em (GOLDBERG, 2003), as informações de identificação pessoal, também chamadas de *Personally Identifiable Information (PII)* são os dados mais sensíveis que um sistema armazena sobre seus usuários. Sistemas de gerenciamento de identidade são os programas desenvolvidos com o objetivo de administrar a coleta, autenticação e uso de identidades e informações ligadas a uma identidade (HANSEN; SCHWARTZ; COOPER, 2008). De acordo com o trabalho (HANSEN et al., 2004), usuários normalmente não estão cientes da disseminação de PIIs durante as transações realizadas quando é utilizado algum tipo de sistema de gerenciamento de identidades.

Assim que os dados dos usuários são enviados a provedores de identidades, o controle sobre como esses dados são divulgados, armazenados e utilizados é perdido (SÁNCHEZ et al., 2012). Porém, usuários como proprietários dos dados armazenados nos provedores, têm o direito de saber quando, por quem e para qual finalidade os seus dados estão sendo utilizados, antes que qualquer processamento ocorra, para

que possam deliberar sobre o consentimento das operações. Assim, é desejável que se mantenha o controle sobre os dados diretamente nas mãos dos usuários (GOLDBERG; WAGNER; BREWER, 1997).

Dados armazenados na nuvem podem ser sensíveis e se ligados à identidade do proprietário podem violar sua privacidade. Assim, este trabalho propõe meios para garantir o controle e privacidade sobre os dados de identificação dos usuários. Também é proposto um meio para prover suporte aos usuários, para que estes sejam capazes de utilizar diferentes graus de liberação de dados no processo de disseminação, visando reduzir os riscos envolvidos com a liberação de dados sensíveis durante esse processo.

Atualmente o tema relatado é abordado de forma pontual, com a aplicação de uso de criptografia, reputação dos provedores da federação e políticas de controle dos dados de modo isolado. Por um lado os trabalhos (SÁNCHEZ et al., 2012) e (BETGÉ-BREZETZ et al., 2012) abordaram o tema de privacidade em provedores em nuvem respectivamente com o uso de reputação e com reputação e criptografia para cifrar dados que são enviados a provedores. Por outro lado os trabalhos (CHADWICK; FATEMA, 2012) e (BETGÉ-BREZETZ et al., 2013) voltaram-se para o uso de políticas de controle de uso de dados. Enquanto que o trabalho (CHADWICK; FATEMA, 2012) utilizou políticas de uso isoladamente para melhorar o controle dos usuários sobre seus dados, o trabalho (BETGÉ-BREZETZ et al., 2013) aplicou o uso de políticas em conjunto com criptografia e reputação das entidades federadas.

Esta dissertação gerou uma publicação inicial (WEINGÄRTNER; WESTPHALL, 2014), onde foram apresentadas as primeiras propostas para os problemas aqui relatados. Este trabalho visa mesclar as diferentes abordagens para prover privacidade e controle aos usuários, utilizando em conjunto políticas de controle de acesso, proposta dos trabalhos (CHADWICK; FATEMA, 2012) e (BETGÉ-BREZETZ et al., 2013) com reputação das entidades federadas e criptografia, proposta com os trabalhos (SÁNCHEZ et al., 2012) e (BETGÉ-BREZETZ et al., 2012).

1.1 JUSTIFICATIVA E MOTIVAÇÃO

Sistemas de gerenciamento de identidades possuem determinados pontos que necessitam ser revisados e repensados antes de serem aplicados em determinados contextos, como no gerenciamento de atributos de identificação de agentes do Estado.

O Estado possui a necessidade de gerenciar de maneira centra-

lizada e organizada as informações de identificação de seus agentes e usuários de sistemas e serviços fornecidos. Porém, concentrar informações de agentes governamentais pode ser um grande atrativo para atacantes, visto que informações sobre presidentes, governadores, senadores entre outros cargos vitais para a soberania nacional estariam concentrados em um único ponto, sujeitos a ataques externos e/ou agentes infiltrados que podem vir a possuir acesso e permissões para as infraestruturas que armazenam esses dados.

Este trabalho está sendo conduzido no âmbito da meta 38 do Laboratório Virtual de Implementações (LATIM) do projeto FINEP/-RENASIC. O LATIM tem como principal objetivo realizar pesquisas em novas técnicas de implementações seguras.

O principal objetivo da Rede Nacional de Segurança da Informação e Criptografia (RENASIC) (RENASIC, 2014) é elevar a competência brasileira em Segurança da Informação e Criptografia. A RENASIC é vinculada ao Centro de Defesa Cibernética (CDCiber), do Comando do Exército. Atualmente a RENASIC trabalha em oito áreas de concentração organizadas em laboratórios, sendo que um deles está associado ao LATIM. A meta 38 do subprojeto LATIM é relacionada ao Gerenciamento de Identidade com Privacidade (GERPRI), onde este trabalho é desenvolvido.

1.2 QUESTÕES DE PESQUISA

Esta seção destina-se a apresentar as questões que iniciaram e motivaram essa pesquisa e dissertação. Três (3) questionamentos principais e já respondidos na bibliografia foram a motivação deste trabalho:

- Usuários são os donos dos atributos de identificação armazenados em IdPs de ambientes federados?
- Devem existir mecanismos que protejam os usuários contra uso não autorizado de PIIs em ambientes federados?
- Deveriam existir meios que forneçam ciência aos usuários quanto ao uso/disseminação de PIIs em ambientes federados?

Estes questionamentos são tratados no capítulo 2 com a apresentação da bibliografia.

Este trabalho busca responder se as ferramentas atuais para criação de ambientes federados contemplam meios para fornecer ciência e controle aos usuários quanto aos PIIs armazenados em IdPs. Além

disso, este trabalho busca propor meios de proteger os PIIs armazenados nos IdPs.

1.3 OBJETIVO GERAL

A proposta deste trabalho visa reduzir os riscos à privacidade dos usuários em sistemas de gerenciamento de identidade. Duas abordagens são previstas para melhorar a privacidade e o suporte dos usuários nos provedores de identidades.

1.3.1 Objetivos específicos

Os objetivos específicos desse trabalho são:

- Propor um modelo para reduzir os riscos à privacidade dos usuários no momento da disseminação de seus PIIs em sistemas de gerenciamento de identidades, provendo suporte ao usuário no processo de disseminação;
- Propor um modelo para prover o controle dos dados de PII aos usuários, de modo que sempre que for requisitado algum dado de identificação, o usuário tenha que estar presente na transação de liberação;
- Estender um sistema de ambiente federado para aplicar as propostas apresentadas neste trabalho.

1.4 METODOLOGIA

A metodologia utilizada para o desenvolvimento desta dissertação está baseada na execução das atividades de levantamento de trabalhos publicados sobre o assunto abordado, análise e discussão desses trabalhos, criação de propostas que solucionem os problemas relatados e por fim a validação das mesmas, comparando-as com propostas de trabalhos anteriores dentro do mesmo contexto. Cada atividade está brevemente descrita a seguir.

- Levantamento de trabalhos relacionados – buscar trabalhos que propõem uma solução parcial ou total para o problema relacionado à privacidade dos dados de identificação dos usuários em sistemas de gerenciamento de identidades;

- Análise – verificação dos trabalhos relacionados, identificar os pontos que podem ser reaproveitados e as deficiências que podem ser exploradas;
- Proposta 1 – tomando como base os pontos fortes dos trabalhos publicados e as deficiências encontradas, propor uma solução que suporte o usuário no momento da disseminação dos dados que utilize a reputação dos provedores para auxiliar os usuários no processo de disseminação, de modo a reduzir os riscos à sua privacidade;
- Proposta 2 – criar um modelo que forneça controle aos usuários sobre os PIIs armazenados nos provedores de identidade;
- Validação da proposta 1 – estender a implementação do protocolo *OpenID Connect*, chamada *MITREid Connect* (MIT, 2014) desenvolvida pelo MIT, para utilizar a reputação dos provedores de serviço para auxiliar os usuários no processo de disseminação de dados;
- Validação da proposta 2 – estender o *framework MITREid Connect* (MIT, 2014), para utilização da proposta de melhoria no processo de gerência de disseminação dos PIIs, para que seja proporcionado ao usuário a possibilidade de gerenciar seus PIIs armazenados cifrados nos provedores de identidades.

Os trabalhos relacionados foram encontrados realizando buscas nas bases de referência bibliográficas:

- IEEE Xplore;
- Science Direct;
- Google Scholar.

Os conjunto palavras chaves de buscas utilizadas foram:

- *privacy and cloud computing federation*;
- *privacy on identity management*;
- *privacy on identity federation*;
- *privacy on identity provider*;
- *privacy and PII.*

1.5 ESCOPO DO TRABALHO

O foco deste trabalho é o controle da disseminação de dados sensíveis em ambientes federados para garantir a privacidade dos atributos de identificação dos usuários, sendo assim algumas questões são desconsideradas para limitar a abrangência das pesquisas. O escopo deste trabalho se limita nos seguintes fatores:

- O modelo aborda aspectos da autenticação, em especial na disseminação dos atributos dos usuários entre provedor de identidades e provedor de serviços;
- O modelo considera que existe um quantificador de reputação, devidamente desenvolvido e implantado;
- O modelo considera que as ferramentas base da implementação como contêiner de aplicação, *frameworks* Spring, jQuery e outros não foram comprometidos e realizam suas tarefas corretamente;
- Apesar de propor o uso de algum método de criptografia para armazenar os PIIs dos usuários o trabalho não realiza uma avaliação profunda de qual método de criptografia seria o mais indicado.

1.6 ORGANIZAÇÃO DO TEXTO

Este trabalho está estruturado da seguinte forma: capítulo 2 descreve os conceitos básicos de computação em nuvem, privacidade, gerenciamento de identidades e ambientes federados; capítulo 3 apresenta os trabalhos relacionados a esta pesquisa; capítulo 4 apresenta o modelo proposto; capítulo 5 apresenta o desenvolvimento e capítulo 6 fecha a dissertação apresentando as conclusões, considerações finais e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta brevemente cada um dos conceitos que são utilizados no decorrer deste trabalho.

2.1 SEGURANÇA NO GERENCIAMENTO DE IDENTIDADE

Esta seção destina-se a apresentar brevemente os conceitos de duas propriedades básicas de segurança e como estas relacionam-se no contexto de gerenciamento de identidades onde a informação mais importante a ser resguardada são as relativas a identidades dos usuários.

A primeira propriedade a ser tratada é a confidencialidade que diz respeito ao controle da informação de modo que entidades e processos não autorizados não façam acesso e uso dessa informação (BERTINO; TAKAHASHI, 2011). Deste modo, em sistemas de gerenciamento de identidades os PIIs devem somente ser disponibilizados para entidades nas quais o usuário explicitamente consentiu acesso.

A segunda, é a integridade, sendo esta voltada a proteger o rigor e exatidão da informação, os dados somente podem ser alterados pelo seu criador e/ou entidades e processos explicitamente autorizados (BERTINO; TAKAHASHI, 2011). Assim, em sistemas de gerenciamento de identidades, somente os usuários ou entidades por eles explicitamente autorizadas podem realizar o processo de edição dos seus PIIs.

2.2 COMPUTAÇÃO EM NUVEM

O conceito de computação em nuvem utilizado por este trabalho é apresentado em (MELL; GRANCE, 2011) como um modelo que permite o acesso ubíquo, conveniente e sob demanda a recursos computacionais, por exemplo: servidores, infraestrutura de rede, armazenamento e aplicações, proporcionando assim, acesso a recursos elásticos com baixo custo de gerenciamento.

Mell e Grance (2011) também definiram três modelos de serviço para fornecimento dos recursos computacionais:

- Software como Serviço – é fornecido ao cliente uma aplicação acessível a partir de vários dispositivos através de interfaces, como um navegador *web* ou uma interface de programa. Os clientes não despendem recursos com manutenção e atualização da infraestru-

tura e aplicação;

- Plataforma como Serviço – modelo em que se provisiona uma plataforma na qual os clientes podem implementar e executar suas aplicações. Nesse modo de fornecimento de recursos os clientes tem controle sobre a estrutura de configuração da plataforma sobre a qual a aplicação é executada, sendo isentos da responsabilidade de manter a infraestrutura necessária para suportar as plataformas. No entanto, o cliente tem que configurar, atualizar e manter a aplicação;

- Infraestrutura como Serviço – os clientes são provisionados com recursos de computação fundamentais, tais como processamento, armazenamento e rede. Nesse modelo o cliente tem o controle sobre essa estrutura virtual, sendo responsável por instalação, configuração e manutenção de sistemas operacionais, *middlewares* e aplicativos, sendo isento da gestão da infraestrutura física, como redes, servidores de processamento e armazenamento e virtualizadores.

Em conjunto com os modelos de serviço, Mell e Grance (2011) apresentaram também os possíveis modelos de implantação e criação de uma nuvem computacional:

- Nuvem privada – toda a infraestrutura de nuvem é direcionada para atender as demandas de um cliente exclusivo. A nuvem é gerida pela organização que está fazendo uso da mesma ou fornecida por um terceiro;

- Nuvem comunitária – a nuvem é provisionada para um grupo de clientes que compartilham políticas e requisitos, sendo gerida pelas organizações que utilizam seus recursos ou fornecida por um terceiro;

- Nuvem pública – a estrutura da nuvem é fornecida para uso do público em geral. É mantida por um terceiro, que vende/aluga os recursos;

- Nuvem híbrida – combinação de dois ou mais modelos de implantação apresentados anteriormente.

2.3 PRIVACIDADE

Landwehr et al. (2012) definem privacidade como o controle da liberação de dados pessoais que os usuários possuem. Complementando, privacidade pode ser definida como um estado em que se está livre de interferências e monitoramento de entidades não desejadas.

Adicionalmente, a Organização das Nações Unidas define em sua declaração universal dos direitos humanos, artigo doze (LAUTERPACHT, 1948) que todos têm direito a privacidade, e que a mesma deve ser garantida pela legislação dos países. Complementando a declaração dos direitos humanos, o conselho de direitos humanos em sua resolução sobre a promoção dos direitos humanos na Internet (COUNCIL, 2012) declara que os mesmos direitos que as pessoas têm off-line também devem ser estendidos ao ambiente on-line.

Em conjunto, tem-se as três principais características encontradas em sistemas de gestão de identidade, em inglês chamados de *Identity Management System (IMS)*, relacionadas com privacidade descritas por Birrell e Schneider (2013):

- *Undetectability* – possibilita ocultar as transações efetuadas pelo usuário, impedindo a detecção das ações de um usuário em um determinado sistema;
- *Unlinkability* – capacidade de ocultar a ligação entre identidade de um usuário e seu histórico de transações;
- *Confidentiality* – provê controle aos usuários sobre a disseminação dos seus atributos.

As propriedades *undetectability, unlinkability* e *confidentiality* são relacionadas, pois definem ações ou métodos para lidar com as partes envolvidas no acesso a um conjunto de dados particular e sensível.

Este trabalho trata a privacidade como sendo um direito fundamental de todo usuário, proporcionando a capacidade dos usuários controlarem a disseminação de seus dados de identificação pessoal no âmbito de sistemas de gerenciamento de identidade, de modo que sejam garantidas as características apresentadas em (BIRRELL; SCHNEIDER, 2013).

A privacidade em sistemas pode ser apresentada sob três diferentes paradigmas (DIAZ; GÜRSES, 2012):

- Privacidade como controle – as violações de privacidade são rotineiramente associadas com a disseminação de dados para terceiros. Assim, as tecnologias voltadas à privacidade proporcionam

meios para os usuários controlarem a disseminação de suas informações e organizações com meios para definirem e forçarem o cumprimento de políticas de segurança que visam prevenir o acesso não autorizado de informações. O principal objetivo desse paradigma é proporcionar aos usuários métodos de monitorar a coleta, processamento e uso de seus dados;

- Privacidade como confidencialidade – o paradigma apresentado anteriormente baseia-se no pressuposto de que as organizações que coletam e processam os dados dos usuários são completamente honestas. No entanto, uma vez que os dados são armazenados em uma organização é difícil para os usuários verificarem como estes estão sendo utilizados. Esse paradigma volta-se para a prevenção de disseminação de informações, concentrando-se em reduzir a quantidade de informações liberadas, de modo a evitar que esses dados possam ser ligados a identidade de um usuário;

- Privacidade como prática social – este paradigma trata a privacidade em seu aspecto social. Usuários normalmente tomam suas decisões de privacidade de acordo com o grupo social ao qual pertencem. Assim, as tecnologias de privacidade nesse contexto buscam deixar os fluxos de dados mais transparentes, proporcionando um melhor entendimento individual e coletivo sobre como as informações são coletadas, analisadas e utilizadas.

As três visões de como abordar privacidade em sistemas apresentados em (DIAZ; GÜRSES, 2012) servem de guia para este trabalho, de modo a prover mais controle, suporte e transparência aos usuários.

2.3.1 Legislação

Existem legislações internacionais e nacionais que visam proteger a privacidade dos usuários no âmbito de sistemas da informação, além da usual que garante o direito dos indivíduos a privacidade fora dos sistemas de informação. Essa subseção destina-se a apresentar as legislações correntes nacionais e internacionais que visam prover direitos aos usuários, de modo a auxiliar à identificação do problema que está sendo abordado.

2.3.1.1 Nacional

No âmbito nacional tem-se a constituição de 1988 (CIVIL, 1988), art. 5º, inciso X, com a declaração de que todo brasileiro tem direito a privacidade. Ainda no art. 5º, inciso XII, é garantida a inviolabilidade das correspondências e das comunicações telegráficas e telefônicas. Porém não há nada específico na constituição quanto ao ambiente da Internet como meio de comunicação e troca de informações.

Buscando sanar a lacuna que existe entre os direitos a privacidade dos brasileiros e o ambiente da Internet, a recém-aprovada lei Nº12.965 de 23 de abril de 2014 (CIVIL, 2014), conhecida como Marco Civil da Internet, estabelece princípios, garantias, direitos e deveres para o uso da Internet no Brasil. O art. 3º, inciso II, declara que a proteção da privacidade deve ser considerada como regra para o uso da Internet no Brasil. Adicionalmente, o art. 7ºdeclara que a Internet é vital para o exercício da cidadania e que usuários devem ser resguardados quanto a sua vida privada, tendo sigilo das comunicações mantido de modo apropriado juntamente com a manutenção de sua conexão com a Internet.

A lei Nº12.965 estabelece também no art. 7º, inciso IX, que todos têm o direito de consentir expressamente sobre a coleta, uso, armazenamento e tratamento de dados pessoais (PIIs) (CIVIL, 2014). Complementarmente, é definida nos artigos 10, 11, 12 e 13 a responsabilidade sobre a guarda e proteção dos registros de conexões dos usuários a Internet, de modo que os provedores não podem terceirizar esse tipo de serviço, devendo respeitar os direitos a privacidade dos usuários estabelecidos anteriormente, armazenando e mantendo os registros de forma segura conforme estabelecido em lei.

2.3.1.2 Internacional

Na Europa existe a diretiva de proteção de dados (DIRECTIVE, 1995), abolindo qualquer processamento de dados referentes ao usuário sem o seu efetivo consentimento. Além disso, a diretiva estabelece que os usuários devem possuir efetivo controle sobre os seus dados de identificação que são coletados e armazenados, sendo que o seu consentimento deve ser requisitado quando houver disseminação e/ou processamento dos dados.

Nos Estados Unidos, existem leis como HIPAA (CONGRESS, 1996), *Gramm-Leach-Bliley Act* (CONGRESS, 1999) e *Children's Online Pri-*

vacy Protection Rule (COMMISSION, 2013) que respectivamente tratam de privacidade e segurança nos sistemas de saúde/telemedicina, sistema financeiro e a coleta e processamento de dados de crianças com idade menor ou igual a 13 anos.

No Canadá há uma lei chamada *Personal Information Protection and Electronic Documents Act* (CANADÁ, 2011), responsável por estabelecer como o setor privado pode coletar, usar e disseminar informações pessoais (PIIs) dos usuários/clientes em suas transações.

2.4 GERENCIAMENTO DE IDENTIDADES

O gerenciamento de identidades pode ser definido como o processo de criação, gerenciamento e utilização de identidades de usuários e a infraestrutura que provê suporte a esse conjunto de processos (LEE; JEUN; JUNG, 2009). Assim, sistemas de gerenciamento de identidades podem ser responsáveis tanto pela tarefa de autenticação quanto pela tarefa da autorização. Quando atuam realizando ambas as funções são chamados de *Identity and Access Management Systems (IAM)*.

Conforme apresentado por Bertino e Takahashi (2011), em um sistema de gerenciamento de identidades existem os seguintes objetos:

- Usuário – entidade que deseja acessar uma aplicação, recurso ou serviço;
- Identidade – conjunto de atributos que representa um usuário, convencionalmente chamados de PII. Quando a identidade não possui todos os atributos que representam o usuário, chamamos de identidade parcial, contendo apenas dados relevantes ao contexto em que o usuário se encontra;
- Provedor de Identidades – também chamado de *Identity Provider (IdP)*, é responsável por fornecer o serviço de gerenciamento de identidades, armazena os atributos que compõem a identidade dos usuários necessários para a utilização dos serviços fornecidos por um provedor de serviços;
- Provedor de Serviços – conhecido em inglês como *Service Provider (SP)*, fornece as aplicações, recursos ou serviços que o usuário efetivamente deseja acessar. O provedor de serviços pode delegar a autenticação dos usuários que acessam seus serviços a um IdP, sendo normalmente encarregado do processo de autorização e deliberação sobre a liberação do recurso requisitado.

Outros autores complementam o conceito de sistemas de gerenciamento de identidade. Chadwick (2009) define gerenciamento de identidades como sendo o processo de gerenciar atributos relacionados a identidades dos usuários. Em contrapartida, Hansen, Schwartz e Cooper (2008) apresentam gerenciamento de identidade como sendo programas e/ou conjunto de ferramentas utilizadas para gerenciar a coleta, autenticação e uso de identidades e informações ligadas a identidades. Assim criam-se meios para os usuários criarem, gerenciarem, e usarem seus atributos.

Este trabalho utiliza o conceito de gerenciamento de identidade como sendo o modelo responsável pela combinação de diferentes módulos que possuem funções como gerência dos atributos que compõem as identidades dos usuários (IdP) e gerência da liberação dos recursos requisitados pelos usuários (SP).

2.4.1 Identidade

Identidade é um conjunto de características/propriedades utilizadas para identificar um sujeito ou organização (BERTINO; TAKAHASHI, 2011). Esse conjunto de informações de identificação pessoal é também chamado de *Personally Identifiable Information* (PII).

Os atributos que compõem uma identidade incluem informações fundamentais para identificar uma entidade e suas preferências em um contexto. Pode-se citar como exemplo de atributos, nome, sobrenome, números de CPF e RG, telefones, e-mails entre outros. A Figura 1 apresenta um exemplo de conjunto de atributos que pode ser utilizado para identificar uma entidade em um determinado contexto.

Atributo	Valor
ID	11111010101
Nome	João da Silva
R.G.	403289440
C.P.F.	362.122.327-41
E-mail	joao.silva@dominio
Telefone	(+55) 48 9999-9999

Figura 1 – Exemplo de uma identidade.

2.4.2 Modelo de controle de acesso

Benantar (2006) descreve modelos de controle de acesso como responsáveis pela avaliação do acesso a um determinado recurso feito por uma identidade comprovada. O recurso pode ser de qualquer natureza como um arquivo em um sistemas de arquivos ou um recurso web como uma *Universal Resource Identifier (URI)*, que pode representar uma função em um sistema.

Existem diversos modelos de controle de acesso conforme apresentado (SMARI; CLEMENTE; LALANDE, 2014); como por exemplo, baseados em confiança, contexto, papéis e atributos. Este trabalho utiliza o modelo de controle de acesso conhecido como *Attribute Based Access Control (ABAC)*, já que este modelo é utilizado em sistemas de gerenciamento de identidades como *Shibboleth* e *OpenId Connect*, sendo o último adotado para a validação das propostas aqui expostas.

No modelo ABAC os atributos são as características ou propriedades de um usuário. Nesse modelo para o processo de autorização são utilizados somente os atributos relevantes ao contexto em que o usuário se encontra, ou seja, é utilizada uma identidade parcial, que representa o usuário no contexto em que ele se encontra. As regras utilizadas para o processo de autorização são aplicadas sobre os atributos que foram obtidos do respectivo IdP em que o usuário está vinculado.

2.4.3 Processos de autenticação e autorização

No gerenciamento de identidades existem dois processos importantes: autenticação e autorização.

O primeiro ocorre convencionalmente nos IdPs com o usuário apresentando suas credenciais que comprovam não somente a existência da identidade, mas também que o usuário é seu detentor. O processo de autenticação é responsável por comprovar a existência e posse de uma identidade por um determinado usuário.

Enquanto que o segundo é executado após o processo de autenticação com sucesso, ocorre normalmente no SP com a aplicação de regras sobre os atributos correspondentes ao usuário advindos do IdP, o processo de autorização é responsável pela deliberação sobre a liberação de acesso a um recurso.

2.4.4 *Login* (SSO) e *Logout* (SLO) únicos

Hursti (1997) discute que conforme o uso de sistemas cresce, principalmente sistemas web, organizações cada vez mais adotam sistemas computacionais em seus processos, cada qual para uma determinada finalidade. Cada sistema possui sua própria gerência de segurança diferente para garantir que somente usuários autenticados e autorizados realizem suas funções. Assim, os usuários necessitam gerenciar diferentes senhas de acesso para diferentes sistemas, além de terem que realizar o procedimento de autenticação apresentando suas credenciais sempre que forem acessar um ou outro sistema. Deste modo, *Single Sign On (SSO)* é útil pois permite que o usuário realize o processo de autenticação uma única vez com uma autoridade de autenticação, sendo que a identidade digital que representa o usuário em um contexto/sistema é repassada da autoridade responsável pela autenticação ao sistema sendo acessado para posterior deliberação sobre a liberação ou não do recurso solicitado.

Adicionalmente, Clercq (2002) apresenta SSO como a habilidade que sistemas fornecem aos usuários para que os mesmos se autentiquem com um autoridade/agente de autenticação uma única vez, para ter acesso a todos os recursos protegidos que eles tem direito sem necessitar realizar o processo de autenticação novamente.

Assim, este trabalho trata SSO como sendo a habilidade dos usuários se registrarem em um único agente (IdP), a partir da autenticação realizada nesse agente os usuários poderiam acessar recursos em diferentes contextos/sistemas (SPs).

Em contrapartida ao SSO, tem-se *Single Logout (SLO)*, ou seja, o processo de desligamento de forma única de um usuário de um ambiente que aplique SSO.

Em ambientes que aplicam SSO, quando o usuário finaliza suas atividades em um dado sistema e deseja sair deste, ao clicar no botão de sair do sistema, ele deveria sair somente da aplicação em questão ou de todas as outras em que ele possa estar autenticado pelo uso de SSO.

Linden e Vilpola (2005) discute que SSO não define uma prática para realizar o desligamento dos usuários do ambiente. Linden e Vilpola (2005) sugere que uma das melhores práticas para sistemas SSO seria o SLO, ou seja, ao invés de cada implementação de SSO ter um comportamento diferente no momento da saída de um usuário do sistema, deve-se seguir a mesma filosofia da autenticação única, ou seja, quando o usuário clicar em um botão de sair de um sistema que use SSO, ele deveria automaticamente sair de todos os outros sistemas que

ele possa estar automaticamente autenticado, essa medida torna o ambiente mais seguro e confiável, sem possibilitar que o usuário saia de um sistema e ainda continue autenticado em outro sem o seu devido conhecimento.

2.4.5 Federação

Chadwick (2009) define federação como uma associação de provedores de serviços e provedores de identidades, através do estabelecimento da confiança entre as partes envolvidas.

Orawiwattanakul et al. (2010) acrescentam que uma federação possibilita que usuários acessem recursos em diferentes domínios administrativos a partir de uma única autenticação (SSO) no domínio ao qual pertencem de modo transparente e seguro.

As entidades em sistema federado dependem umas das outras para autenticar usuários e prover acesso aos serviços, através de protocolos de comunicação padronizados. Assim, organizações são capazes de compartilhar aplicações sem necessidade de adotar as mesmas tecnologias de segurança, autenticação e autorização.

Este trabalho considera federação de identidades como sendo um conjunto de padrões e tecnologias que permite o compartilhamento seguro e transparente de identidades entre diversas organizações. Como exemplo de federações tem-se:

- CAFe – Desenvolvida pelo grupo de trabalho Rede Nacional de ensino e Pesquisa (RNP). É uma federação de identidade que reúne instituições brasileiras e utiliza Shibboleth como ferramenta para criação do ambiente, teve seu início em 2007 (CAFE, 2007).

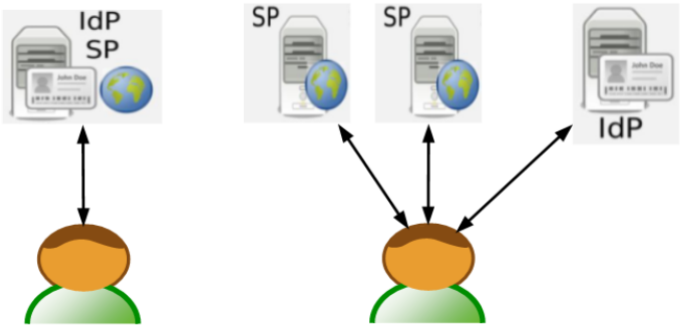
- SWITCHaai – Criada em 2004 é construída pela comunidade SWITCH para acesso a diversos sistemas de *e-learning* e aplicações web para a comunidade universitária suíça, utiliza o sistema Shibboleth como ferramenta para criação da federação (SWITCH, 2004);

- Google products – Pode-se considerar a suíte de produtos fornecida pela empresa Google como sendo uma federação, dado que é utilizado um IdP centralizado que permite que não somente seja feita a autenticação de um usuário com os sistemas por ele fornecido, mas também pode-se utilizar os dados armazenados em seu IdP para autenticar usuários em outros sistemas, com o uso do protocolo *OpenId Connect* (GOOGLE, 2014);

2.4.6 Modelos de sistemas de gerenciamento de identidades

O gerenciamento de identidades pode ocorrer tanto em um único domínio administrativo, também conhecido como *Administrative Domain (AD)*, ou em um ambiente com múltiplos ADs, como uma federação.

Conforme Jøsang e Pope (2005) apresentam, o gerenciamento das identidades dos usuários pode ocorrer internamente no SP ou externamente em uma aplicação dedicada. A Figura 2(a) ilustra o gerenciamento de identidades em um AD único ocorrendo internamente no SP. Enquanto que a Figura 2(b) demonstra uma aplicação dedicada (IdP) responsável pela autenticação e fornecimento de dados para o processo de autorização no SP.



(a) Embutido. (b) Aplicação dedicada.

Figura 2 – Gerenciamento de identidades em um AD único.

Jøsang e Pope (2005) também apresentam o funcionamento do gerenciamento de identidade em um ambiente com múltiplos domínios administrativos (ADs), em que é possível ter vários IdPs fornecendo autenticação e dados para o processo de autorização em diversos SPs pertencentes a diferentes domínios. Assim, as organizações podem cooperar entre si, agregando vários serviços de parceiros. Conforme ilustrado na Figura 3, a identidade de um usuário de um domínio é válida para acessar serviços em diferentes ADs.

Como apresentado na Figura 3, o domínio A e B cooperam entre si e formam uma federação, deste modo usuários do domínio A (Bob) é capaz de acessar serviços no domínio B de forma transparente, realizando o processo de autenticação no seu domínio de origem (A).

As interações apresentadas na Figura 3 disparadas por Bob são ilustradas com setas sólidas, enquanto que as interações disparadas e realizadas por Alice são apresentadas com as setas tracejadas.

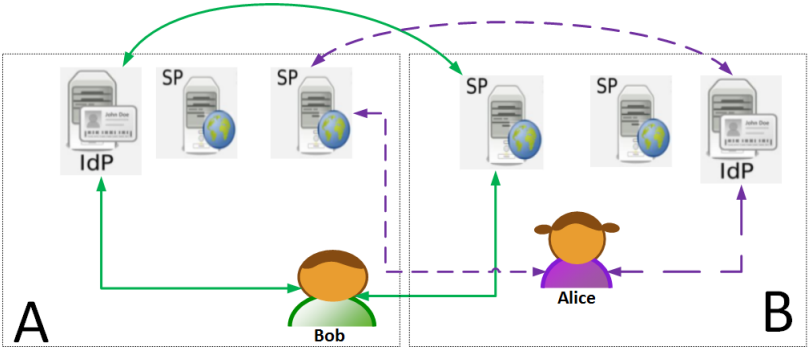


Figura 3 – Gerenciamento de identidades federado (múltiplos ADs).

2.4.7 Ferramentas de gerenciamento de identidades

Esta seção destina-se a apresentar as ferramentas de gerenciamento de identidade para ambientes federados de modo que suas características sejam descritas e comparadas para posterior decisão e escolha da ferramenta que melhor se enquadra no contexto deste trabalho.

2.4.7.1 Authentic 2

Desenvolvido pela empresa francesa Entr'ouvert (ENTR'OUVERT, 2014), o projeto é distribuído sem custos e possui seu código fonte aberto sob a licença GNU GPL.

É desenvolvido em *Python*, fazem uso de um framework chamado Django como base para criação da ferramenta. Além do uso de *Python*, fazem uso da linguagem *C* em módulos que necessitam de velocidade, como por exemplo, criptografia.

Possui suporte para múltiplos protocolos de troca de mensagens entre IdP e SP: SAML 2.0, CAS 1.0, CAS 2.0, OpenID 1.0, OpenID 2.0. Para efetuar o processo de autenticação são suportados os se-

guintes mecanismos: OAUTH, certificados digitais, autenticação com *OpenId* e SAML 2.0 em provedores de terceiros, LDAP e banco de dados (padrão de usuário e senha). Além disso, pode ser utilizado como *proxy*, convertendo um protocolo em outro, como por exemplo convertendo mensagens em SAML 2.0 para o padrão OpenID.

A ferramenta Authentic 2 é basicamente um provedor de identidades (IdP) sendo que é sugerido para a implementação do provedor de serviços e integração com o IdP(authentic 2) o uso da biblioteca Lasso no lado do SP, para gerenciar as mensagens trocadas entre IdP, SP e usuários.

Pode-se destacar sobre o projeto:

- Desenvolvido com tecnologia de código aberto (*open source*);
- Criado e mantido por uma empresa ao invés de uma fundação sem fins lucrativos;
- O módulo conversor de padrões parece ser promissor por facilitar a integração de diferentes ambientes, padrões e tecnologias;
- Pouco difundido, poucas organizações fazem uso dessa tecnologia.

2.4.7.2 Higgins (*Personal Data Service*)

Criado e mantido pela fundação Eclipse, é um modelo para gerenciamento, armazenamento e uso de atributos ligados a identidade dos usuários (ECLIPSE, 2014).

É utilizado um *plugin* no navegador do usuário que verifica cada página acessada e determina se o mesmo está autenticado no serviço, se ele não estiver o *plugin* realiza o processo de autenticação no serviço, preenchendo automaticamente os campos necessários para efetuar o *login* com os dados do usuário.

Possui um servidor que armazena os dados (atributos dos usuários) desenvolvido em Java e pode ser instalado e configurado localmente para o usuário, em uma máquina virtual na nuvem ou acessado como serviço fornecido por terceiros. O *plugin* desenvolvido e instalado no navegador verifica a necessidade de login e de apresentar dados dos usuários nos sistemas web acessados, assim para preencher os formulários apresentados ao usuário o sistema busca informações armazenadas no seu módulo servidor.

É um *framework* de código aberto para a integração de identidades, perfis e informação distribuída, que utiliza o paradigma de

gerenciamento de identidade centrada no usuário, mantendo cartões de identificação com informações de contexto e credenciais dos usuários. Os cartões de identificação são identidades parciais criadas para representar o usuário em um determinado contexto/sistemas.

Tem a vantagem de utilizar o padrão SAML que é amplamente difundido no ambiente acadêmico, porém não se enquadra na proposta deste trabalho por ser um *framework* com uma arquitetura diferenciada e que necessita da utilização de *plugins* do lado do cliente, assim como extensões do navegador.

2.4.7.3 OpenAM

Criado pela Sun como uma solução para SSO, conhecido como OpenSSO, após a aquisição da mesma pela empresa Oracle, o projeto OpenSSO foi transformado em produto (OracleSSO) e descontinuado. Um grupo de usuários e desenvolvedores do OpenSSO fundou então uma empresa chamada ForgeRock e acolheu o código que estava aberto do openSSO, criando assim o projeto openAM (FORGEROCK, 2014). O projeto é distribuído sob a licença *Common Development and Distribution License (CDDL)* - 1.0. Dessa forma, seu código é aberto para a comunidade, e seu uso é gratuito, sendo somente cobrado o suporte fornecido pela empresa ForgeRocks, caso se queira suporte profissional para a ferramenta.

Possui o módulo provedor de identidades (desenvolvido em Java) chamado de OpenAm, responsável por realizar o processo de autenticação e por decidir se um usuário pode ou não acessar um recurso. Seu módulo OpenAM pode cooperar com os padrões SAML, OpenId Connect e Oauth2, para obter integração com provedores de serviço de terceiros, provendo SSO e SLO a estes sistemas.

No recurso que está sendo protegido é necessário o uso de um agente chamado de *Policy Agent* que intercepta toda requisição ao recurso e direciona o usuário para a autenticação, além de realizar o processo de autorização, verificando se o usuário está apto a acessar o endereço de localização do recurso, conhecido como *Universal Resource Locator (URL)*. O módulo agente possui duas implementações: uma desenvolvida na linguagem de programação *C* para um *webserver* como o *Apache Web Server* e outra em Java para contêineres como *JBoss* e *Tomcat*.

A Figura 4 ilustra o fluxo de troca de mensagens nos processos de autorização e autenticação do usuário utilizando a ferramenta OpenAM

descrito na documentação da própria ferramenta (FORGEROCK, 2011). Os passos 6 e 8 são realizados sem a interação dos usuários; aplicação cliente e servidor OpenAM trocam mensagens diretamente um com o outro.

1. Usuário solicita um recurso protegido;
2. Agente intercepta a requisição e envia o usuário para realizar o processo de autenticação no servidor OpenAM;
3. Servidor OpenAM apresenta ao usuário sua tela de *login*, onde o mesmo deve entrar suas credenciais;
4. Usuário informa suas credenciais ao servidor OpenAM;
5. Após a validação das credenciais e comprovação da identidade, gera-se uma prova de autenticação (*ticket*) que deve ser apresentado ao Agente no recurso protegido;
6. Agente valida o *ticket* recebido junto ao servidor OpenAM, durante esse processo é verificado com a aplicação de políticas se o usuário possui permissão de acesso ao recurso solicitado;
7. Avaliação das políticas do usuário;
8. Envio de resposta ao agente sobre o questionamento do usuário estar ou não autorizado para acessar o recurso;
9. Com base na resposta do passo anterior, deliberação sobre a entrega do recurso ao usuário.

Deste modo, após ter apresentado o projeto, suas características e funcionamento pode-se destacar os seguintes pontos:

- Desenvolvido com tecnologia *open source*;
- Criado e mantido por uma empresa ao invés de uma fundação sem fins lucrativos;
- Assim como o projeto *Authentic* 2, seu módulo conversor de padrões parece ser promissor por facilitar a integração de diferentes ambientes, padrões e tecnologias;
- Possui um fluxo de autorização de acesso a recurso um pouco diferente do que normalmente se utiliza no âmbito de federações, centralizando as requisições em seu servidor OpenAM que realiza

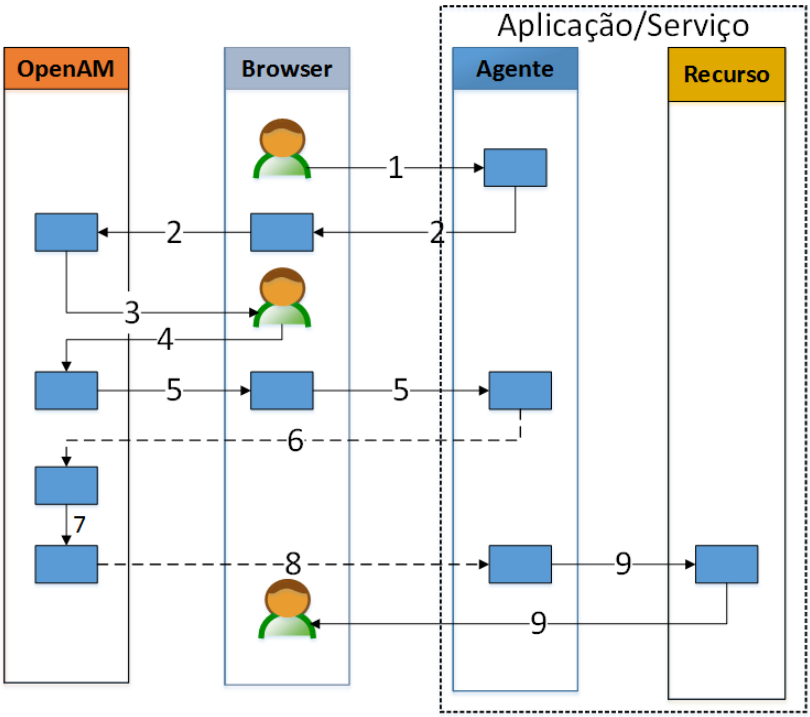


Figura 4 – OpenAM - autorização e autenticação
(FORGEROCK, 2011)

o processo de autenticação/autorização. Isso pode dificultar a escalabilidade quando o contexto são aplicações para nuvem em que deve-se considerar a natureza elástica e dinâmica do ambiente.

2.4.7.4 OpenId connect

Criado e mantido pela fundação OpenID como uma evolução das tecnologias existentes e tentativa de unificação dos diversos protocolos para integração de serviços na web. *OpenId Connect* (OPENID, 2014c) é um protocolo para trocas de dados de identificação dos usuários entre SP e IdP desenvolvido sobre o protocolo OAuth 2.0.

Sua especificação contempla uma *Application Programming Interface (API)* para *Web Services* que seguem o padrão *Representa-*

tional State Transfer (REST), possibilitando a integração de diferentes dispositivos e tecnologias. Assim como em outros ambientes de federação possui módulos (protocolos) para troca dos dados de identificação, serviço de descoberta de provedores, registro dinâmico de provedores e gerenciamento das sessões estabelecidas pelos usuários (SSO e SLO).

Possui ambas as implementações servidor IdP e cliente (*framework* para aplicação provedora de serviço utilizar seu protocolo) em diversas linguagens como *Java, C, PHP, Python e Ruby.*

É um protocolo para estabelecer relações de federação com os mais diferentes provedores de identidade e serviço, sendo independente de plataforma e tecnologia. Possui diversas implementações desenvolvidas por projetos que fazem uso da mesma (OPENID, 2014a): *MIT, Ping Identity, Google, Apache, Microsoft* e *JBoss.*

No protocolo *OpenID Connect* por questões de compatibilidade com a especificação OAUTH 2.0 a nomenclatura usa o termo aplicação cliente como sinônimo de SP e servidor de autorização como sinônimo de IdP.

A Figura 5 apresenta o fluxo de troca de mensagens para realizar o processo de autenticação, transmissão dos atributos do usuário e liberação do recurso protegido, conforme descrito em sua especificação (OPENID, 2014b).

Na Figura 5, os passos 2 e 5 aparecem em dois momentos por representarem uma requisição *HTTP Redirect*, sendo composta da solicitação do IdP ou SP ao navegador do usuário e posteriormente a execução dessa solicitação pelo navegador. No passo 2 o SP solicita que o usuário se autentique em um IdP. Enquanto que no passo 5, após o processo de autenticação ser realizado com sucesso, o IdP solicita que o usuário acesse o serviço desejado com um *token* que possibilita a obtenção de dados para a avaliação de liberação ou não do recurso pelo SP.

1. Usuário requisita um recurso protegido;

2. A aplicação cliente deve criar uma solicitação de autenticação para o usuário se autenticar no servidor de autorização;

3. Usuário é autenticado com a apresentação de suas credenciais ao IdP;

4. Após a autenticação, ocorre a obtenção do consenso do usuário para liberação dos atributos necessários à aplicação cliente;

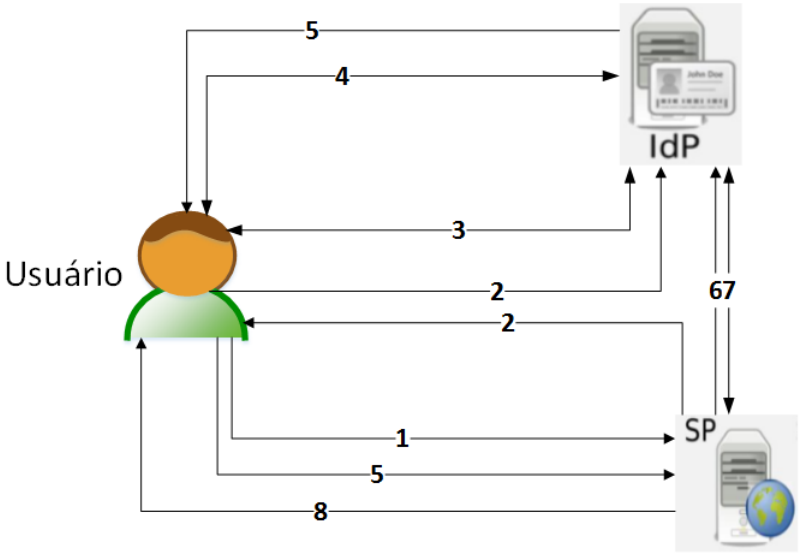


Figura 5 – OpenID Connect Core 1.0

5. O IdP envia o *ticket* gerado para liberação do recurso para o SP;

6. O SP solicita uma resposta quanto ao *ticket* ao IdP, assim obtém-se um identificador de usuário e um *token* de acesso a dados. Este passo somente confirma a autenticação do usuário no IdP, sem a obtenção de dados sobre esse usuário;

7. O SP pode obter dados do usuário caso necessário, além do identificador obtido, através da apresentação do *token* de acesso ao servidor de autorização (passo opcional). Somente são liberados nesse processo os dados que o usuário autorizou a serem compartilhados do IdP ao SP;

8. Liberação do recurso.

É uma ferramenta que se destina a realizar a integração de diferentes provedores (IdP e SP) nos mais diferentes dispositivos e nas mais diversas tecnologias. É mantido pela fundação OpenID, sendo ela responsável por especificar os componentes existentes nesse protocolo. A documentação é bem detalhada e de fácil compreensão, assim como suas implementações. É utilizado por organizações como Google, MIT, Ping Identity, entre outros.

2.4.7.5 Ping Federate

É um produto criado pela empresa Ping Identity focado em prover um ambiente de federação completo para as empresas (IDENTITY, 2014).

Possui a capacidade de cooperar com os padrões OAuth, OpenID Connect e WS-Trust. A empresa afirma facilitar a aplicação do conceito de federação, proporcionando mais segurança no controle de acesso dos usuários as aplicações que estão utilizando os seus serviços.

O sistema permite desde SSO único até SSO Federado, desenvolvido para suportar padrões abertos como SAML e OpenID, conseguindo assim, realizar integração com diferentes tecnologias. Seus esforços são para simplificar a integração com diferentes aplicações, atuando como mediador de autenticação entre usuários e aplicações.

É uma infraestrutura de código fechado que é vendida, aparentemente utiliza a especificação OpenID Connect, sendo que por ser fechada torna-se inviável o seu uso na elaboração de melhorias por este trabalho

2.4.7.6 Shibboleth

O Shibboleth (SHIBBOLETH, 2014) foi criado e é mantido pela organização Internet 2, baseado no padrão SAML para troca de mensagens entre cliente, SP e IdP de informações relevantes a identidade e realização dos processos de autenticação e autorização.

É uma ferramenta para criação de ambientes federados de código aberto possuindo o módulo provedor de serviços (SP), que gerencia a segurança do recurso protegido, módulo provedor de identidade (IdP) e de descoberta de provedores chamado *Discovery Service (DS)*. O modulo IdP é somente desenvolvido em Java, enquanto que o SP foi criado utilizando a linguagem de programação *C* como um módulo para o *Apache Web server*, para que assim como no OpenAM possa proteger um recurso.

A Figura 6 ilustra o fluxo de troca de mensagens para realizar a gerência de segurança do recurso e processo de autenticação e autorização com o uso do Shibboleth:

1. O usuário acessa o recurso protegido utilizando um navegador;
2. direciona-se o usuário ao módulo DS, onde é apresentada uma série de IdPs, sendo que o usuário deve selecionar o IdP em que

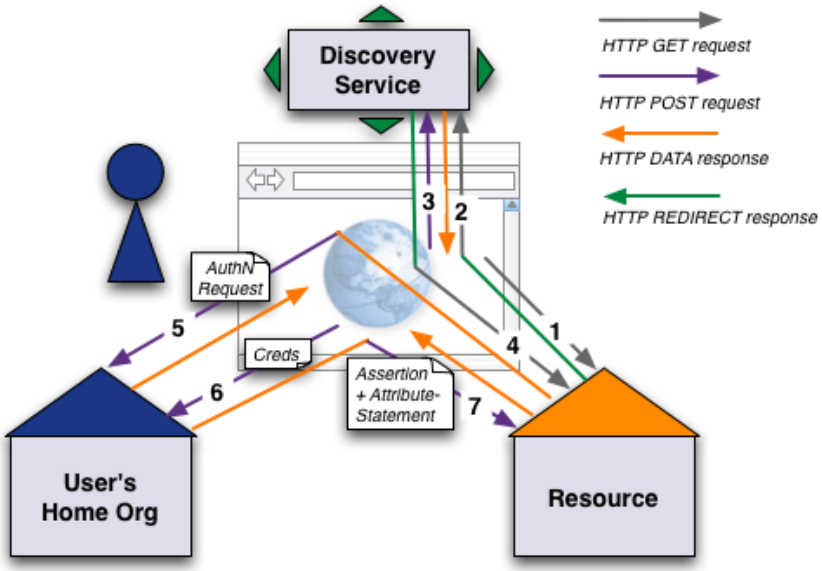


Figura 6 – Shibboleth
(SWITCH, 2014)

ele possui cadastro para prosseguir;

3. O usuário informa no módulo DS qual IdP ele está vinculado;

4. no DS é gerada uma requisição para iniciação de sessão no recurso protegido, e o usuário é redirecionado ao seu IdP para autenticação;

5. O IdP responde a solicitação de autenticação com uma página de login, na qual o usuário deve apresentar suas credenciais;

6. O usuário apresenta suas credenciais que são checadas pelo IdP gerando assim uma asserção com os atributos do usuário;

7. A asserção criada é então enviada ao recurso protegido através do navegador do usuário;

8. Com os dados no recurso protegido, o recurso pode enfim ser liberado ao usuário.

O Shibboleth é muito utilizado no meio acadêmico, a seu favor tem o código aberto, sendo mantido por uma fundação.

Porém, não possui uma especificação formalizada sobre a implementação. A documentação é incompleta dificultando a sua configuração e entendimento do seu funcionamento. Adicionalmente, possui somente uma implementação em *C* para o Apache Web Server para atuar como agente de segurança no recurso a ser protegido, limitando e dificultando a sua aplicação em determinados casos, como aplicações desenvolvidas em outras linguagens e que não utilizem o Apache Web Server. Outro problema encontrado é de que o ambiente criado com o Shibboleth não suporta SLO nativamente, necessitando algumas modificações e extensões das aplicações padrões.

3 TRABALHOS RELACIONADOS

Neste capítulo são descritos os trabalhos relacionados. Os trabalhos serão descritos e discutidos de modo a indicar os problemas a serem sanados quanto à privacidade no âmbito do gerenciamento de identidades tanto em ambiente de um único AD (ambientes web simples) ou de múltiplos ADs como em ambientes federados e nuvens computacionais.

Existe um *plugin* para o componente provedor de identidades (IdP) do ambiente de federação Shibboleth, *uApprove* (SWITCH, 2012). O *plugin* melhora o processo de disseminação de informações, deixando os usuários cientes de quais dados são enviados do IdP para o SP, conforme ilustrado na Figura 7. Porém, os usuários não podem customizar o processo, isto é, não conseguem escolher quais PIIs podem ou não ser disseminados.

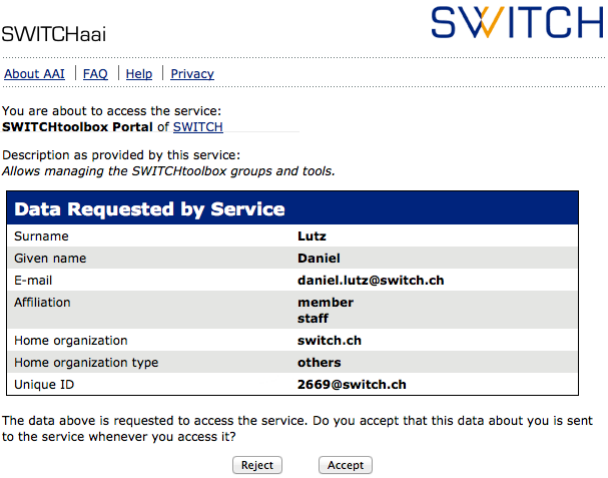


Figura 7 – uApprove, User Consent Module for Shibboleth Identity Providers

(SWITCH, 2012)

De forma semelhante a (SWITCH, 2012), Orawiwattanakul et al. (2010) abordaram a falta de transparência na troca de informações de identificação dos usuários entre IdPs e SPs em ambientes federados. É proposto um modelo pelo qual o usuário, após a realização do processo de autenticação e antes de acessar o serviço, teria como verificar

quais atributos relacionados à sua identidade estão sendo enviados ao SP, possibilitando a seleção dos atributos que o usuário deseja ou não compartilhar com o SP, conforme apresentado na Figura 8. Assim, foi desenvolvida uma extensão para o módulo IdP do Shibboleth que proporciona aos usuários um melhor controle sobre os dados de identificações que são armazenados nos sistemas.

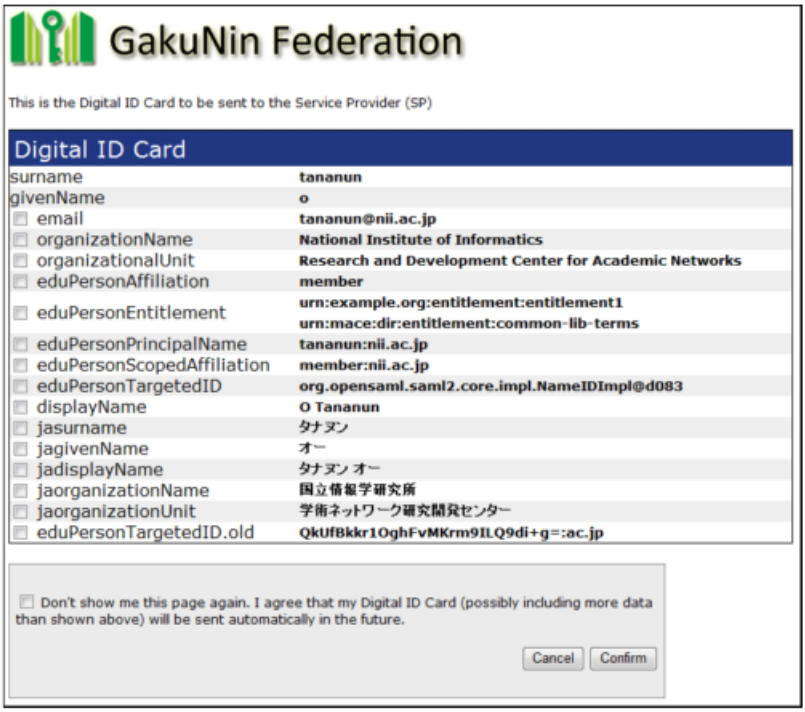


Figura 8 – Disseminação de dados customizável com o uApproveJP (ORAWIWATTANAKUL et al., 2010)

Conforme apresentado na Figura 8 o usuário em questão , sabendo que os dados exigidos pelo SP são o sobrenome (*surname*) e o nome (*givenName*), poderia optar por acessar o SP sem enviar nenhum dado extra. Porém, para obter a customização do serviço de acordo com seu perfil, ele poderia enviar, por exemplo, o seu e-mail e o nome da organização para qual ele trabalha, podendo assim obter serviços diferenciados.

Com uma abordagem diferente das anteriores, Paredes e Zorzo

(2012) apresentaram uma abordagem que proporciona privacidade para serviços já consolidados e fornecidos como o *Google App Engine*. O modelo faz uso de uma aplicação intermediária entre o cliente e o serviço utilizado. Assim todo dado que sai desse ponto intermediário para a nuvem seria cifrado e todo dado que sai da nuvem passando por esse ponto seria decifrado para o cliente. O trabalho (PAREDES; ZORZO, 2012) garante que ninguém no provedor da nuvem(provedor de espaço de armazenamento como por exemplo: *Dropbox, Google Drive* e *Microsoft one drive*) se aproveitaria dos dados dos usuários, por estes estarem cifrados. No entanto, na camada proposta não foi apresentado qualquer mecanismo que evitasse vazamentos e utilização indevida dos dados que ali trafegam. Administradores da camada proposta poderiam possuir meios de obter indevidamente informações dos usuários.

Sánchez et al. (2012) propuseram um modelo voltado a privacidade e segurança em federações dinâmicas. É utilizado um protocolo de reputação que possibilita a mensuração da reputação dos participantes da federação. Usuários teriam a habilidade de checar o nível de reputação de um determinado SP com os demais membros da federação para então decidir se enviam ou não determinados dados para este provedor. Assim, o usuário teria a possibilidade de definir quais tipos de informações podem ser enviadas para quais SPs de acordo com sua reputação. O modelo prevê meios para que os usuários consultem como estão sendo utilizados os dados enviados aos SPs, e com base nessa informação, pode-se aumentar ou diminuir a reputação de um SP. Porém, se um SP tem alta reputação não significa que o mesmo não está vulnerável a ataques (externos e internos) e vazamentos de dados.

Betgé-Brezetz et al. (2012) buscaram sanar os problemas de privacidade e segurança que os usuários enfrentam quando enviam dados para os provedores de nuvem com o uso de níveis de confiança. Foi proposto um meio pelo qual usuários poderiam definir o quanto eles confiam em um determinado SP, com base nesse nível de confiança, o usuário poderia enviar os dados para a nuvem de três modos distintos:

- Dado aberto – dados enviadas sem nenhum mecanismo de criptografia;
- Dado parcialmente criptografado – as informações são enviadas cifradas em conjunto com metadados sobre o dado em aberto;
- Dado completamente cifrado – os dados são enviados totalmente criptografados sem nenhum tipo de metadado ou informação extra em aberto.

No trabalho (BETGÉ-BREZETZ et al., 2012) também é proposto um pacote chamado de *Privacy Data Envelope (PDE)*, que serviria como meio de transporte para os dados em um dos três modos de envio apresentados anteriormente. O PDE também teria a possibilidade de carregar em conjunto com os dados, políticas de uso. Essas políticas poderiam definir quem, como, onde e quando os dados podem ser utilizados e quais ações podem ser realizadas com os mesmos.

Chadwick e Fatema (2012) discutem sobre a falta de meios para se definir políticas de acesso e uso dos dados na nuvem e padrões para se aplicar tais políticas. Assim, foi proposto um modelo composto por uma série de serviços web, *Web Services (WS)*, que seriam locais com a capacidade de avaliar as políticas definidas para um determinado dado, retornando uma resposta sobre a ação que se pretende executar. Toda aplicação/sistema ao receber o dado de um cliente iria obter em conjunto uma série de políticas de uso. Deste modo, antes de executar qualquer ação (por exemplo, copiar, ler, apagar, mover), a aplicação deveria consultar um dos WSs responsáveis por avaliar as políticas que foram definidas e somente prosseguir com a ação solicitada, caso o WS retorne com uma resposta positiva para a ação solicitada.

De acordo com a proposta apresentada em (CHADWICK; FATEMA, 2012), os WSs utilizados para analisar as regras e decidir se uma aplicação pode ou não realizar um processamento não provê garantias de aplicação das políticas definidas e respeito a privacidade dos usuários, visto que não foram apresentados meios de garantir a utilização da estrutura de WS proposta por parte da aplicação no lado do SP. Adicionalmente, mesmo que a aplicação faça uso dos WSs, não foi apresentado um meio para fazer com que ela respeite o retorno do WS consultado.

Combinando o uso de políticas de acesso a dados apresentado em (CHADWICK; FATEMA, 2012) com o uso de reputação dos trabalhos (SÁNCHEZ et al., 2012; BETGÉ-BREZETZ et al., 2012), Betgé-Brezetz et al. (2013) apresentam um modelo que faz uso do pacote PDE em conjunto com políticas de uso de dados, visando garantir privacidade e segurança entre usuários e SPs. No modelo proposto o PDE carrega os dados em conjunto com suas políticas de uso, definidas pelo usuário, para o SP que está na nuvem. De acordo com o modelo, haverá no SP um componente chamado de *Data Protection Module (DPM)*, que seria responsável por avaliar as políticas de uso dos dados antes de realizar qualquer ação com os dados. Também foi estabelecido que o PDE contendo os dados e as políticas somente poderia ser enviado para os nós da nuvem que contenham DPMs instalados para a avaliação das

políticas.

Porém, o trabalho apresentado em (BETGÉ-BREZETZ et al., 2013) possui o mesmo problema encontrado em (SÁNCHEZ et al., 2012; BETGÉ-BREZETZ et al., 2012); boa reputação não significa que o SP vai obedecer as políticas definidas por seus usuários e que não é vulnerável a administradores de sistemas curiosos/maliciosos que possuem meios e habilidade técnica para se apossar indevidamente dos dados dos usuários. Os usuários não possuem meios para verificar se os módulos DPM foram desenvolvidos, implantados e estão funcionando corretamente.

Os trabalhos apresentados que utilizam reputação e políticas podem ser classificados no paradigma de privacidade como controle apresentado por (DIAZ; GÜRSES, 2012) e discutido previamente na seção 2.3. Esses trabalhos consideram que as organizações vão ser completamente honestas quanto à manipulação dos dados dos usuários que são coletados, armazenados e processados. Os trabalhos que fazem uso de criptografia abrangem o paradigma de privacidade como confidencialidade como apresentado por (DIAZ; GÜRSES, 2012). Sob o aspecto social, utilizando a noção de grupo para deliberar sobre a disseminação de dados os trabalhos que usam reputação fazem uso desse paradigma, tendo em vista que as decisões para liberação ou não de atributos são realizadas com base em uma métrica de reputação que é obtida com a utilização de opiniões de outros usuários e membros da federação.

A Tabela 1 apresenta os trabalhos previamente descritos e discutidos com suas respectivas características. Os trabalhos foram classificados quanto a:

- Uso de criptografia (UC) – aplicação ou não de criptografia para cifrar dados armazenados nos provedores (IdPs e/ou SPs);
- Utilização de reputação (UR) – mensurar a reputação dos provedores para deliberar sobre o envio de determinados dados a um provedor;
- Políticas de uso/disseminação (PUD) – utilização de políticas de acesso e uso dos dados enviados aos provedores, que deveriam ser avaliadas pelos mesmos antes de executar qualquer tarefa com os dados;
- Notificação de uso/disseminação (NUD) – existência de meios pelo quais o usuário é notificado sobre o uso e disseminação dos seus dados, especialmente os dados de PII;
- Suporte no processo de disseminação (SPD) – disponibilizações de meios pelos quais os usuários recebem suporte no processo de

disseminação, auxiliando na escolha de quais grupos de atributos são mais adequados para o SP acessado. Deste modo, tenta-se evitar a liberação indiscriminada de dados de identificação a provedores de serviços que podem vir a ferir a privacidade dos usuários.

Tabela 1 – Características dos trabalhos que abordam privacidade dos dados enviados aos provedores em nuvem e/ou federação.

Trabalhos	Características				
	UC	UR	PUD	NUD	SPD
(SWITCH, 2012)				X	
(ORAWIWATTANAKUL et al., 2010)			X	X	
(PAREDES; ZORZO, 2012)	X				
(SÁNCHEZ et al., 2012)		X			
(BETGÉ-BREZETZ et al., 2012)		X			
(CHADWICK; FATEMA, 2012)			X		
(BETGÉ-BREZETZ et al., 2013)	X	X	X		
Esta dissertação	X	X	X	X	X

Nenhum dos trabalhos apresentados aborda completamente todas as propriedades aqui discutidas. Organizações estão sujeitas falhas de segurança e ataques, o que pode gerar danos aos usuários caso ocorra vazamento de suas informações de identificação. Desta forma, não basta usar apenas um paradigma de tratamento de privacidade, deve-se considerar uma junção dos paradigmas discutidos, de modo a aprimorar a privacidade dos usuários.

4 REDUZINDO OS RISCOS A PRIVACIDADE EM SISTEMAS DE GERENCIAMENTO DE IDENTIDADE

Este capítulo destina-se a apresentar uma abstração do modelo atual dos sistemas federados e a discutir o problema relacionado à privacidade dos atributos de identificação dos usuários que são armazenados em provedores de identidades. Também é apresentado um modelo para solucionar os problemas elencados durante o processo de discussão e análise da bibliografia.

4.1 MODELO ATUAL DE GERENCIAMENTO DE IDENTIDADES

Os fluxos das trocas de mensagens das ferramentas descritas na seção 2.4.7 possuem diversas semelhanças. Assim, um modelo abstrato foi criado para guiar o desenvolvimento e extensão de aplicações para o gerenciamento de identidades. Deste modo, este trabalho apresenta o modelo atual de gerenciamento de identidades que será estendido para aplicação das propostas aqui expostas.

Os passos a seguir descrevem o fluxo normal de autenticação e troca de atributos entre uma aplicação provedora de serviço e outra provedora de atributos de identificação do usuário, quando um usuário acessa um recurso protegido sem estar autenticado no sistema, de acordo com a ilustração da Figura 9:

1. O usuário requisita um recurso protegido ao SP – não existe um contexto de segurança na aplicação com os atributos do usuário, assim é necessário solicitar a autenticação do mesmo para obtenção dos atributos e criação do contexto de segurança;

2. O provedor de serviço (SP) solicita a autenticação do usuário – a aplicação SP deve criar uma solicitação de autenticação para o usuário efetuar o processo de autenticação no seu respectivo provedor de identidades (IdP), essa solicitação deve redirecionar o usuário ao IdP;

3. Comprovação da identidade – usuário apresenta suas credenciais ao IdP;

4. Obtenção do consenso – após o processo de autenticação ter finalizado com sucesso, ocorre a obtenção do consenso do usuário para liberação dos atributos necessários ao SP;

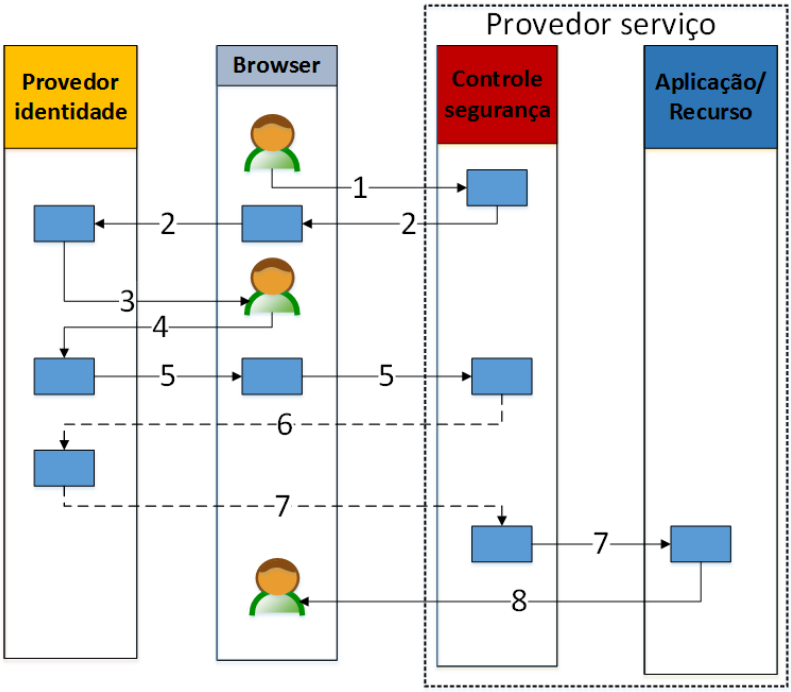


Figura 9 – Modelo Genérico de troca de mensagens em um IMS

5. Enviar confirmação de autenticação ao SP – criar uma mensagem ou *ticket* que confirme o processo de autenticação ocorrido no IdP para o SP, após a geração da confirmação, deve-se direcionar o usuário com a confirmação ao SP;

6. Validação da confirmação de autenticação – SP solicita uma resposta quanto a confirmação de autenticação recebida, assim obtém-se um identificador de usuário e um *token* de acesso a dados extra, que podem ser obtidos caso necessário;

7. Obtenção de dados adicionais (opcional) – o SP pode obter dados adicionais do usuário quando necessário, através da apresentação do *token* de acesso ao IdP;

8. Liberação do recurso requisitado pelo usuário.

Apesar da descrição dos passos se assemelhar com o protocolo *OpenId Connect*, ele pode ser utilizado tanto para descrever o funcio-

namento em ambientes que fazem uso do *Shibboleth* quanto *OpenAM*, mesmo alguns passos sendo descritos de uma forma diferente, o processo a ser realizado é em essência o mesmo. O passo 4 não ocorre nativamente em ambientes que usam Shibboleth, porém com uso de extensões como uApprove (SWITCH, 2012) e uApprove.jp (ORAWIWATTANAKUL et al., 2010) ele passa a ter o mesmo comportamento.

4.2 ESCOLHA DE UMA PLATAFORMA DE FEDERAÇÃO

A Tabela 2 apresenta um comparativo das ferramentas para criação de ambientes federados descritas anteriormente. Assim, o projeto OpenID Connect foi escolhido para servir como base de extensão e aplicação das propostas, por ser de código aberto, mantido por uma fundação, possuir um protocolo formalizado que especifica o padrão de troca de mensagens e formato das mensagens.

As outras ferramentas foram desconsiderados pelos motivos apresentados na sequência. Os projetos Higgins e Authentic 2 possuem baixa adesão tanto pela indústria quanto pelo meio acadêmico, o que afeta a comunidade que atua no âmbito desses projetos, tornando sua evolução e manutenção lenta e complicada. Os projetos OpenAM e Ping Federate foram descartados por pertencerem a empresas com fins lucrativos, apesar disso o OpenAM possui código fonte aberto, enquanto que o Ping Federate é fechado. Mesmo com a grande adesão da comunidade acadêmica o projeto Shibboleth foi desconsiderado por não possuir especificação relatando o seu funcionamento, que força o desenvolvedor/pesquisador a estudar profundamente o seu código/implementação.

4.3 RISCOS À PRIVACIDADE EM SISTEMAS DE GERENCIAMENTO DE IDENTIDADE

Existem legislações e padrões para tratamento dos dados em sistemas de informação focados em estabelecer diretrizes para proteger os direitos dos usuários. Das legislações, pode-se citar as apresentadas anteriormente: (DIRECTIVE, 1995; CONGRESS, 1996, 1999; COMMISSION, 2013; CANADÁ, 2011; CIVIL, 2014).

Dentre alguns dos padrões que visam mitigar os problemas em torno da privacidade e segurança dos usuários em sistemas de informação, pode-se citar: o guia de proteção da privacidade no fluxo de dados entre países da OECD (OECD, 2013), o guia de segurança e privacidade

Tabela 2 – Características das ferramentas de federação

Projeto	Código aberto	Possui especificação?	Proprietário	Padrões	Adesão
Shibboleth	Sim	Não	Internet2	SAML 1/2	Academia
OpenAM	Sim	Não	ForgeRock	SAML 1/2, OpenId Connect	Indústria
Authentic 2	Sim	Não	Entr'ouvert	SAML 2, OpenID 1/2	Baixa
OpenID Connect	Sim	Sim	OpenID	OpenId Connect (Json)	Indústria/ Academia
Higgins	Sim	Não	Eclipse	SAML	Baixa
Ping Federate	Não	Não	PingIdentity	SAML, OpenID	Indústria

do NIST (JANSEN; GRANCE et al., 2011) e o guia da *Cloud Security Alliance* (ALLIANCE, 2011).

Existe também um esforço para tentar reduzir os riscos de violação de privacidade dos dados que são armazenados fora dos domínios dos usuários, com os trabalhos apresentados anteriormente: (SWITCH, 2012; ORAWIWATTANAKUL et al., 2010; HAN-ZHANG; LIU-SHENG, 2010; PAREDES; ZORZO, 2012; SÁNCHEZ et al., 2012; BETGÉ-BREZETZ et al., 2012; CHADWICK; FATEMA, 2012; BETGÉ-BREZETZ et al., 2013). Porém, mesmo assim, ainda há problemas a serem resolvidos no quesito de privacidade no gerenciamento de identidades:

- Falta de controle sobre PIIs do usuário – usuários não possuem meios eficazes para gerenciar seus dados de identificação que são armazenados nos provedores de identidade. Conforme discutido em (SÁNCHEZ et al., 2012), uma vez que dados como PIIs são enviados para sistemas de gerenciamento de identidade, o controle sobre como o dado está sendo armazenado, utilizado e compartilhado é perdido;

- Processo de disseminação de PIIs pouco transparente – os usuários devem estar cientes sobre a liberação dos dados de PII ao acessar um SP. Conforme abordado no trabalho (ORAWIWATTANAKUL et al., 2010), usuários como detentores dos dados tem o direito de saber a finalidade da coleta dos dados para então deliberar sobre a aceitação do processo de disseminação. Essa característica apesar de importante é pouco explorada, como pode ser constatado

na Tabela 1, onde são comparados os trabalhos relacionados;

- Falta de suporte no processo de disseminação de PIIs – Zhang et al. (2007) relataram que usuários não conseguem definir com êxito as suas políticas de disseminação de informações pessoais, enquanto que Hansen, Schwartz e Cooper (2008) discutiram que uma única configuração padrão para disseminação de atributos de usuários não vai atender as necessidades de todos. Desta forma, usuários deveriam ser capazes de customizar os provedores de identidades para definir de modo granular quais atributos podem ser enviados para cada tipo de SP.

A Fundação *Open Security* descreve que o volume de incidentes de vazamento de atributos de identificação se mantem alto ao longo dos anos com uma certa variação para mais ou menos (OSF, 2014), conforme ilustrado com a Figura 10. Adicionalmente, mais de um terço dos ataques são consequências de agentes internos, sejam eles acidentais ou intencionais (OSF, 2014), apresentado na Figura 11.

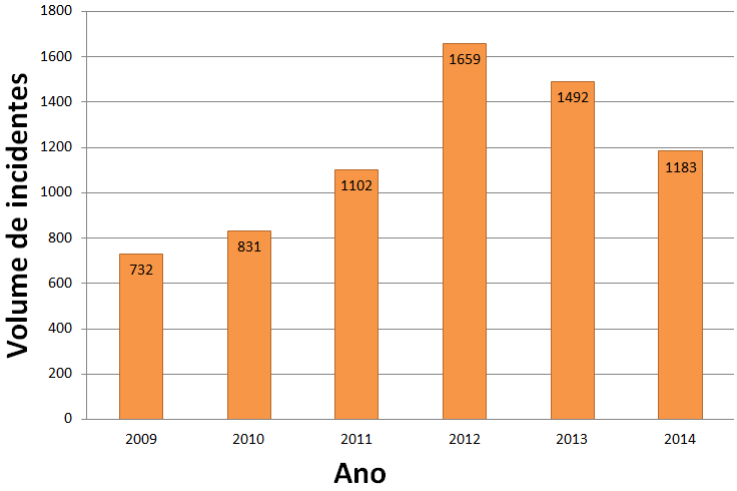


Figura 10 – Volume de incidentes ao longo dos anos.

Como discutido anteriormente nos trabalhos (HAN-ZHANG; LIU-SHENG, 2010; CHADWICK; FATEMA, 2012; BETGÉ-BREZETZ et al., 2013; XIAO; XIAO, 2013), podem existir entidades curiosas/maliciosas internas aos provedores (IdPs e SPs) com privilégios e habilidades técnicas para violar a privacidade dos usuários. Este fato pode ser observado

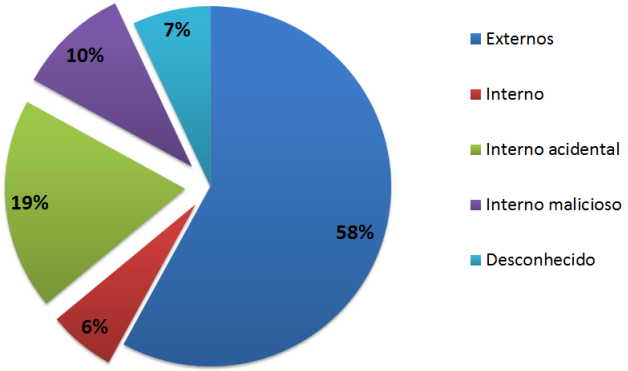


Figura 11 – Origem dos ataques que resultaram os incidentes contabilizados.

nas Figuras 10 e 11 com os dados sobre volume de incidentes e fontes de ataques.

Usuários como donos dos atributos de identificação que são armazenados nos provedores de identidades têm o direito de saber quando e por quem estes dados estão sendo utilizados e para qual propósito, como já estabelecido em diversas legislações (DIRECTIVE, 1995; CONGRESS, 1996; CANADÁ, 2011; CIVIL, 2014). Ao enviarem seus atributos de identificação para provedores de identidades de modo a facilitar o uso de serviços federados, os usuários acabam perdendo controle desses dados. Desse modo, ambientes federados podem ferir a privacidade dos usuários, caso uma entidade maliciosa dentro do IdP se aproprie indevidamente dos atributos de identificação dos usuários.

4.4 MODELAGEM DAS AMEAÇAS AOS PIIS DOS USUÁRIOS NOS IDPS

O conhecimento das ameaças é vital para a definição dos requisitos de um sistema, pois auxilia na seleção das medidas de segurança que serão desenvolvidas, implementadas e implantadas (DENG et al., 2011).

Uma modelagem de ameaças é uma abordagem estruturada para identificar potenciais vulnerabilidades de um sistema que podem vir a ser exploradas. Essa modelagem tem como foco determinar o atacante e como este poderia ter sucesso na sua tarefa. É uma forma de antecipar as falhas que poderiam afetar o sistema (STEVEN, 2010).

Uma das metodologias utilizadas para a modelagem de ameaças é conhecida como STRIDE e desenvolvida pela empresa Microsoft (HERNAN et al., 2006). Entretanto, essa metodologia não cobre ameaças relacionadas à privacidade. Deste modo, este trabalho utilizou como base o trabalho (DENG et al., 2011), que adiciona quesitos de privacidade na metodologia STRIDE.

Assim, antes de qualquer proposta de melhoria na privacidade dos sistemas de gerenciamento de identidade, era necessário o conhecimento da aplicação IdP em detalhes, os recursos que devem ser resguardados, as entidades e suas interações com a aplicação, as diferentes dependências de terceiros e pontos de recebimento de requisições que a aplicação possui. Com esse levantamento, foi desenhado um Diagrama de Fluxo de Dados (DFD) da aplicação e as árvores de ameaças para os problemas discutidos anteriormente, bem como suas causas e consequências.

4.4.1 Descrição da aplicação modelada (IdP)

Os provedores de identidades são componentes fundamentais de sistemas de gerenciamento de identidades. São responsáveis pela gerência dos dados dos usuários, armazenamento, atualização, autenticação e disseminação de atributos relevantes a identidades dos usuários.

Os atributos de identificação dos usuários são sensíveis e somente devem ser atualizados, inseridos e removidos do sistema pelo próprio usuário. Nem mesmo os administradores dos sistemas dependentes podem possuir acesso a esses dados.

4.4.2 Dependências de aplicativos de terceiros

Esta subseção lista as dependências de aplicativos externos que a aplicação IdP possui. As ferramentas aqui descritas são de fundamental importância para a aplicação, todas as dependências aqui descritas são mantidas seguras com aplicação recorrente de pacotes de atualização conforme política de gerência da organização. O gerenciamento dos servidores também ocorrerá conforme políticas padrões de gerência da organização:

- Sistema operacional – utilizado para gerenciar os recursos de hardware que são fornecidos para a aplicação;
- Contêiner de aplicações web – fornece uma estrutura básica de

funções para a aplicação realizar seus trabalhos. Pode ser utilizado em diferentes tipos de sistemas operacionais;

- Banco de dados – utilizado para o armazenamento das informações relevantes aos usuários, pode ser de qualquer versão, sendo executados em diferentes sistemas operacionais;

- Redes virtuais (VLANs) – a conexão entre o banco de dados e a aplicação será efetuada em uma rede privada e isolada para tal finalidade;

- Utilização do protocolo TLS/SSL – Toda comunicação entre cliente e aplicação e aplicação e banco de dados se dará por um canal de comunicação seguro (TLS/SSL);

- Utilização do protocolo HTTPS - O IdP somente atenderá requisições utilizando o protocolo *Hypertext Transfer Protocol (HTTP)* sobre um canal de comunicação seguro (TLS/SSL), ou seja, utilizará o protocolo *Hypertext Transfer Protocol Secure (HTTPS)* para troca de mensagens com seus clientes.

4.4.3 Agentes que interagem com a aplicação IdP

Entidades que possuem algum envolvimento com a aplicação ou sua estrutura de suporte (servidores de aplicação, rede e sistemas operacionais). São listadas aqui as entidades e seus respectivos papéis e interações com o IdP.

1. Usuário não autenticado – somente recebe a página de autenticação, não podendo realizar nenhuma função no sistema. Pode acessar a página de cadastro;

2. Usuário autenticado – usuário que possui dados armazenados no provedor de identidade e acessa serviços providos por terceiros com a utilização desses atributos. Esse usuário pode disseminar seus atributos a outros serviços, bem como gerenciar, inserir, atualizar e remover dados do sistema;

3. Desenvolvedor da aplicação (IdP) – realiza o desenvolvimento da aplicação, possui conhecimento sobre o modelo de dados, métodos de acesso ao banco e permissões de acesso;

4. Administrador do banco de dados – administrador do banco de dados realiza as manutenções e atualizações. Possui acesso de leitura e escrita no banco;

5. Administrador do Servidor de aplicação – efetua a implantação da aplicação, configura o servidor de aplicação, realiza a aplicação de atualizações e atualizações da aplicação em produção;

6. Administrador da infraestrutura – realiza a manutenção nos sistemas operacionais dos servidores, possui acesso de leitura e escrita nos discos e servidores de armazenamento. Efetua a atualização e manutenção preventiva nos servidores;

7. Aplicação externa (SP) – aplicação que requisita um usuário a efetuar o processo de autenticação e espera algum atributo do mesmo para liberar o acesso a um recurso/serviço.

4.4.4 Pontos do IdP para recebimento de requisições

Nesta subseção são detalhados todos os pontos do IdP que atendem a requisições e as respectivas entidades que se relacionam com os mesmos.

Os pontos para tratamento de requisições na aplicação IdP são:

- Função de autenticação – página utilizada pelos usuários para informarem suas credenciais;

- Função de gerenciamento de PIIs – página utilizada para gerenciar os dados (PIIs) cadastrados no IdP;

- Função de cadastro – página utilizada pelos usuários para inserirem seus dados no IdP, permitindo posterior autenticação dos mesmos;

- Função comprovação de autenticação – valida o comprovante de autenticação gerado após uma autenticação ser realizada. Atende a requisição de validação realizada pelo SP;

- Função de disseminação de PIIs – Realização o envio dos dados necessários (PIIs) para o SP.

A tabela 3 mapeia os pontos da aplicação IdP que realizam alguma interação com agentes externos.

Tabela 3 – Mapeamento dos pontos de interação com os respectivos agentes

Ponto de interação	**Agentes de interação**
Função de autenticação	Usuário não autenticado
Função de gerenciamento de PIIs	Usuário autenticado
Função de cadastro	Usuário não autenticado
Função comprovação de autenticação	Aplicação externa (SP)
Função de disseminação de PIIs	Usuário autenticado, aplicação externa (SP)

4.4.5 Recursos a serem protegidos/mantidos

Esta subseção destina-se a elencar e descrever os recursos que a aplicação possui e devem ser devidamente segurados e garantidos, de modo que somente as entidades relacionadas com os mesmos tenham acesso.

Os recursos podem ser divididos em dois grupos: os que dizem respeito aos atributos dos usuários (PIIs) e os que são relevantes para manutenção de funcionamento e disponibilidade da aplicação IdP.

Quanto aos recursos pertinentes aos usuários têm-se:

- Credenciais de autenticação – dados utilizados para realização do processo de comprovação de posse de uma identidade;
- Atributos de identificação (PIIs) – informações que podem ser utilizadas para identificar um usuário e suas preferências em um determinado contexto.

Quanto aos recursos pertinentes a aplicação IdP têm-se:

- Funções/serviços – atividades/procedimentos realizados pela aplicação para realização de suas tarefas;
- Disponibilidade – deve ser garantida e mantida a disponibilidade da aplicação;
- Acesso ao repositório de dados – a aplicação IdP necessita de permissão de leitura e escrita no repositório de dados para realizar suas tarefas.

A tabela 4 mapeia os recursos da aplicação IdP que devem ser resguardados com agentes que realizam algum tipo de interação com os mesmos.

Tabela 4 – Mapeamento dos recursos resguardados com seus respectivos agentes

Recursos resguardados	**Agentes de interação**
Credenciais de autenticação	Usuário autenticado
Atributos de identificação (PIIs)	Usuário autenticado
Funções/serviços	Usuário autenticado, usuário não autenticado, aplicação externa (SP) e desenvolvedor da aplicação (IdP)
Disponibilidade	Desenvolvedor da aplicação (IdP), administrador do servidor de aplicação e administrador da infraestrutura
Acesso ao repositório de dados	Administrador do servidor de aplicação, administrador da infraestrutura e administrador do banco de dados

4.4.6 DFD das interações da aplicação IdP

Esta subseção destina-se a apresentar as camadas de atuação de cada um dos agentes descritos anteriormente na aplicação avaliada. A Figura 12 apresenta um DFD da aplicação IdP, onde são ilustradas as interações entre as entidades existentes no contexto de seu gerenciamento e uso.

O conjunto de administradores e desenvolvedores tem seu limite de interações definida pela linha tracejada em vermelho da Figura 12, detendo acesso e controle quanto aos recursos de disponibilidade, repositório de dados e funções/tarefas executadas pela aplicação IdP. Têm a responsabilidade de garantir esses recursos e para tal possuem acesso de super usuário. Porém, devem ser limitados quanto ao acesso a informações armazenadas no repositório de dados, como por exemplo, dados pessoais dos usuários do sistema, de modo que seja mantida a privacidade quanto aos dados (PIIs) que estão armazenados no banco de dados.

O conjunto de usuários é composto por usuário (usuário devidamente cadastrado e autenticado) e usuário não autenticado, possuindo um limite de interações definido pela linha tracejada em verde apresentada na Figura 12. Os usuários cadastrados e autenticados no sistema podem alterar e disseminar seus respectivos dados de identificação armazenados no IdP, possibilitando assim que os mesmos acessem serviços

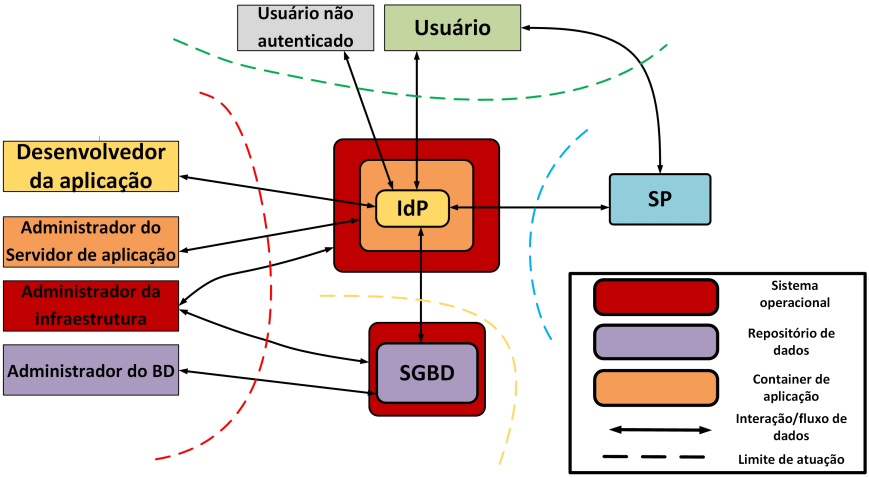


Figura 12 – DFD dos pontos de interação dos agentes com a aplicação IdP.

fornecidos por SPs. Os usuários que não possuem cadastro no sistema podem somente realizar o processo de cadastro, inserção de dados de identificação no sistema IdP, para que então possam se autenticar no sistema e acessar serviços e recursos.

O limite do IdP está marcado pelo tracejado em amarelo ilustrado na Figura 12, estabelecendo comunicação com o repositório de dados em que se encontram os atributos dos usuários. As suas interações se restringem em acessar o repositório para os processos de autenticação, gerência dos atributos (realizado pelos usuários autenticados) e cadastro de atributos (realizado pelos usuários ainda não cadastrados e não autenticados).

O SP possui as interações marcadas pelo tracejado em azul apresentado na Figura 12, realizando interações para validar o comprovante de autenticação e obter os dados que o usuário concordou em compartilhar com o mesmo.

4.4.7 Árvores de ameaças

Com o DFD da aplicação IdP construído, criou-se as árvores de ameaças com o intuito de mapear as ameaças que o sistema está sujeito e que foram tratadas.

As árvores de ameaça seguem a seguinte estrutura, o primeiro elemento representa o problema, os elementos intermediários são as causas, os fatos que geram o problema, enquanto que o elemento da base são as consequências que o problema descrito pode vir a causar.

A Figura 13 apresenta a ameaça a privacidade dos PIIs, causada pela falta de ciência dos usuários sobre quais atributos estão sendo disseminados no momento que estes acessam um provedor de serviços. Essa ameaça é causada por falta de transparência no processo de disseminação de atributos entre IdP e SP.

Figura 13 – Ameaça 1 – Falta de ciência na disseminação de atributos.

Além da ameaça causada pela falta de transparência no processo de disseminação, há a falta de mecanismos para garantir o controle dos atributos dos usuários que lá estão armazenados. A Figura 14 demonstra a ameaça de carência no controle dos atributos que pode ser causada por um dos seguintes fatores:

- Controle de acesso falho/fraco no Banco de Dados (BD) – um usuário/administrador do BD, caso possua acesso à tabela que detêm os dados dos usuários poderia se apropriar desses dados indevidamente sem a ciência dos usuários;
- Controle de acesso falho/fraco no servidor do BD – o administrador do servidor que executa o BD ou então um atacante que tenha

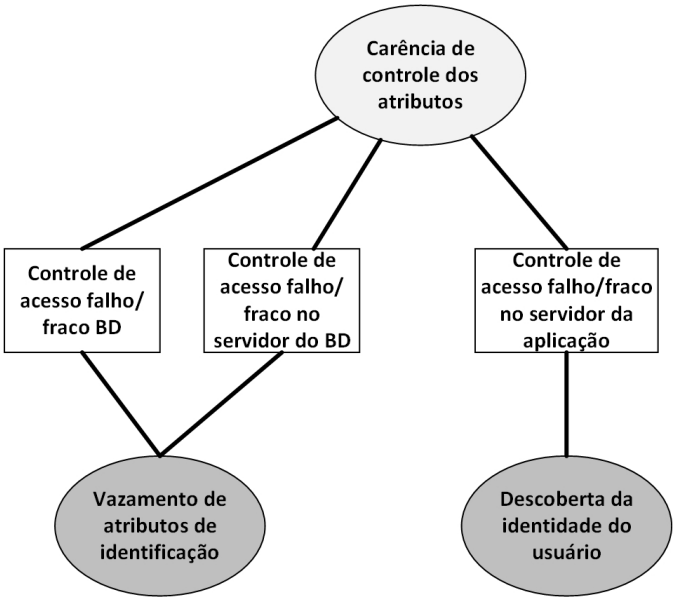


Figura 14 – Ameaça 2 – Carência no controle dos atributos dos usuários.

conseguido acesso ao mesmo, poderia se apossar dos arquivos do BD que estão no sistema de arquivos, montar esses arquivos em outro local e apossar-se dos dados que ali estão armazenados;

- Controle de acesso falho/fraco no servidor da aplicação (IdP) – caso o administrador do servidor (infraestrutura) e/ou administrador do servidor de aplicação que hospeda a aplicação inspecionem dos dados da memória do servidor, possíveis atributos do usuário que estivessem armazenados em memória para uma transação poderiam ser descobertos e obtidos por entidades curiosas/maliciosas.

A última ameaça encontrada quanto a privacidade dos usuários no IdP, é apresentada na Figura 15. É causada pelo processo de disseminação não proporcionar customização e suporte aos usuários. Essa ameaça pode ser causada pelos seguintes fatores:

- Falta de suporte no processo de disseminação – os usuários na maioria dos casos não sabem quais dados se liberados podem proporcionar maior risco a sua privacidade. Deste modo, sem

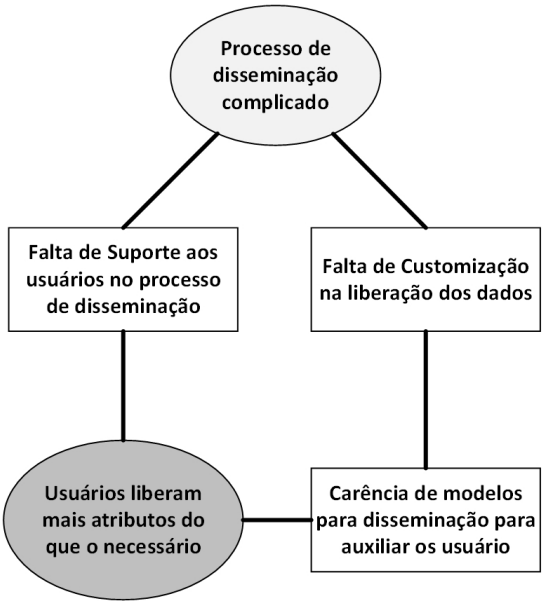


Figura 15 – Ameaça 3 – Processo de disseminação complicado e sem possibilidade de customização.

um suporte adequado, eles podem acabar liberando um determinado conjunto de atributos que pode vir a por em risco a sua privacidade;

- Falta de customização do processo de liberação de dados – o processo não permite customização para uma liberação granular dos atributos, o que pode não satisfazer a necessidade de todos os usuários, levando-os a liberar mais do que o necessário/desejado.

Com base nas árvores de ameaça e da descrição do modelo de sistemas federados apresentadas anteriormente, as ameaças à privacidade dos usuários em sistemas de gerenciamento de identidades em ambientes federados em computação em nuvem que esta dissertação pretende tratar são descritos a seguir.

- Ameaça 1, falta de ciência na disseminação de atributos. Seguindo a discussão da ameaça apresentada com a Figura 13 em conjunto com os passos de troca de mensagem entre IdP-SP ilustrados na Figura 9, realização do processo de número 4. Detectou-se que usuários desses sistemas usualmente não estão cientes de

quais atributos são disseminados a seu respeito. Porém, conforme abordado no trabalho (ORAWIWATTANAKUL et al., 2010), usuários como detentores dos dados tem o direito de saber a finalidade da coleta dos dados e quais dados estão sendo liberados para então deliberar sobre a aceitação do processo de disseminação;

- Ameaça 2, carência no controle dos atributos dos usuários. Conforme discutido com a Figura 14 em conjunto com a discussão realizada em (SÁNCHEZ et al., 2012), uma vez que dados como PIIs são enviados para sistemas de gerenciamento de identidade atuais, o controle sobre como ele está sendo armazenado, utilizado e compartilhado é perdido;

- Ameaça 3, processo de disseminação complicado e sem possibilidade de customização. Como discutido com a Figura 15, usuários sofrem com a falta de suporte para deliberar quais dados podem ser disseminados e quais podem vir a causar menor ou maior dano a sua privacidade. Zhang et al. (2007) relatam que usuários não conseguem definir com êxito as suas políticas de disseminação de informações pessoais, enquanto que Hansen, Schwartz e Cooper (2008) discutiram que uma única configuração padrão para disseminação de atributos de usuários não vai atender as necessidades de todos. Desta forma, usuários deveriam ser capazes de configurar os provedores de identidades com modelos para definir quais atributos podem ser enviados para cada tipo de SP.

4.5 MODELO PARA TRATAR A PRIVACIDADE EM SISTEMAS DE GERENCIAMENTO DE IDENTIDADE

A proposta deste trabalho altera o modelo tradicional de gerenciamento de identidades apresentado na seção 4.1, de modo que sejam reduzidos os riscos quanto a privacidade dos usuários com seus dados de identificação armazenados nos IdPs.

Para facilitar o entendimento e apresentação o modelo foi fragmentado em duas partes, a primeira relativa ao tratamento das ameaças 1 e 2 sendo a segunda parte dedicada ao tratamento da ameaça 3.

4.5.1 Abordando controle e ciência dos usuários nos IdPs

Esta subseção destina-se a apresentar a proposta com as medidas para tratar as ameaças 1 e 2, respectivamente descritas e discutidas com as Figuras 13 e 14. De modo a tratar as ameaças 1 e 2 utilizou-se de forma combinada as propostas de criptografia de dados enviados e provedores na nuvem proposta por (PAREDES; ZORZO, 2012) e (BETGÉ-BREZETZ et al., 2013) com as propostas de ciências e controle do usuário apresentadas em (SWITCH, 2012) e (ORAWIWATTANAKUL et al., 2010).

Para que usuários acessem serviços federados é necessário realizar o processo de cadastro em um provedor de identidade, que valida de alguma forma o cadastro, realiza o processo de autenticação e disseminação dos dados dos usuários para provedores de serviços. Os processos de autenticação e disseminação são apresentados com a Figura 9. O acesso a um serviço qualquer em um sistema federado é ilustrado com as interações em azul na Figura 16.

Este trabalho propõe que usuários cifrem seus dados antes dos mesmos serem armazenados nos IdPs. Assim, como foi discutido anteriormente, o usuário passa a ter um controle mais apurado sobre os dados que estão armazenados nos IdPs, evitando que agentes internos aos IdPs façam uso dos mesmos sem prévio consenso e conhecimento do usuário.

O processo de cadastro no modelo proposto é representado pelas interações verdes na Figura 16, os usuários (Bob e Alice) ao inserirem os seus dados de identificação nos IdPs realizarão o processo de codificação dos mesmos com uma chave, esse processo ocorrerá no lado do cliente (navegador) antes do envio dos dados ao IdP. Com isto cria-se uma camada de proteção no dado e dificulta-se o uso sem a participação do usuário na transação.

Atualmente é prevista a cifragem dos dados dos usuários de dois modos:

- Chave do usuário – utilizando-se uma chave pública de um par de chaves que o usuário possui, como por exemplo, a chave pública de um certificado digital;
- Palavra chave – os usuários podem informar um texto, um conjunto de bits que é utilizado para derivar um par de chaves, do qual obtém-se a primeira para cifrar os dados antes que os mesmos sejam enviados ao IdP.

Os passos realizados para enviar os dados cifrados do usuário ao IdP são ilustrados no fluxograma da Figura 17, os passos em verde são

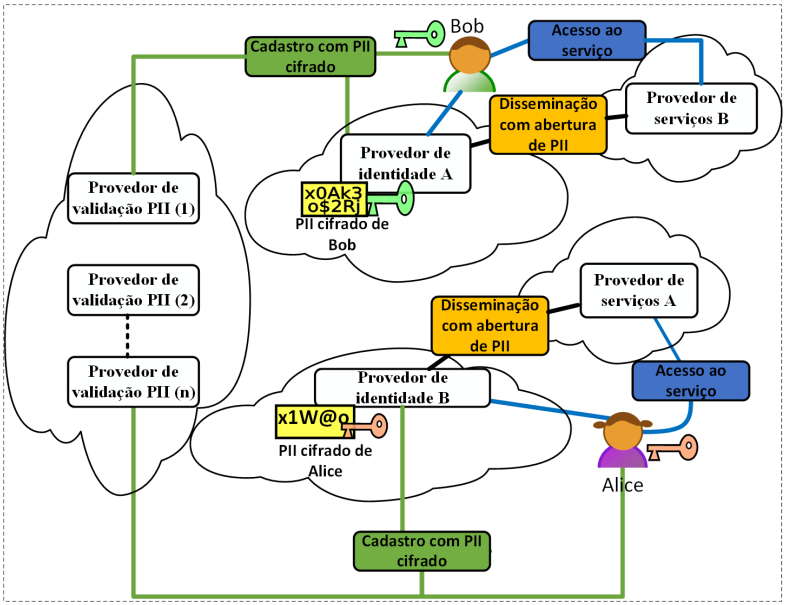


Figura 16 – Proposta para tratamento das ameaças 1 e 2.

realizados pelo usuário, enquanto que os azuis são realizados por uma biblioteca de criptografia executada no cliente e os passos em amarelo são executados na aplicação IdP ao receber a requisição de solicitação de cadastro com os dados do usuário.

Os processos apresentados no diagrama da Figura 17 são detalhados a seguir:

- Para acessar serviços de um ambiente federado o usuário precisa estar vinculado a um provedor de identidades. Deste modo, o primeiro passo para o usuário é o cadastro em um IdP;
- Após acessar a página de cadastro o usuário entra com seus dados;
- Decide se vai ser utilizado uma palavra chave para cifrar os dados ou então uma chave de um certificado digital que ele já possui;
- Entra com a chave para a biblioteca fazer o processo de codificação;
- A biblioteca verifica se é necessário gerar um par de chaves, caso seja, gera-se o par e obtém-se a primeira chave do par. Caso não

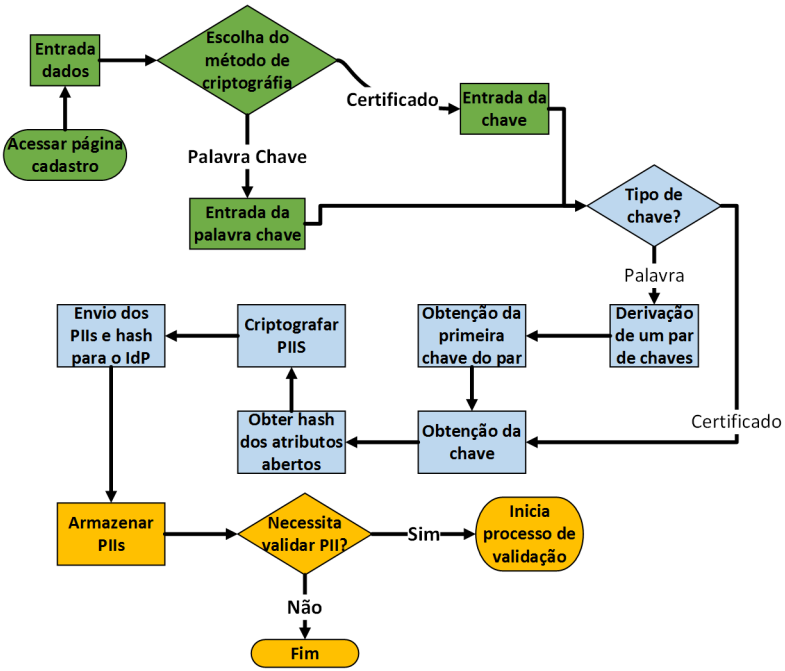


Figura 17 – Cadastro dos dados com privacidade.

seja necessário a biblioteca obtém a chave do certificado informada pelo usuário;

- Obtém o *hash* de cada um dos atributos do usuário para ser enviado em conjunto com o dado cifrado para o IdP. Esse dado é utilizado pelo SP para verificação de conformidade dos atributos armazenados com os atributos disseminados;

- De posse da chave a biblioteca realiza a codificação dos dados (o nome utilizado para fazer o login e a senha não são cifrados);

- É realizado o envio do conjunto de atributos e *hash* dos mesmos para o IdP para realizar o processo de armazenamento;

- IdP recebe os dados e verifica a necessidade de validar os PIIs;

- Caso seja necessário a validação, inicia-se o processo de validação, para então armazenar o dado.

A proposta de armazenar atributos cifrados nos provedores de identidades não é necessariamente por não confiar na entidade, até porque caso não haja confiança não haveria federação. Porém, mesmo confiando na entidade não é possível confiar em todos os seus funcionários e mesmo uma entidade confiável está sujeita a falhas em sua segurança.

O armazenamento de atributos cifrados impossibilita para o IdP validar e garantir a veracidade dos dados que armazena. Assim, é proposto o uso de provedores de serviços especializados em validação dos atributos. Na federação existirão provedores que somente seriam responsáveis por validar um determinado atributo que por si só não iria ferir a privacidade dos usuários, mas garantiriam a veracidade do dado cifrado armazenado no IdP.

Pode-se ter por exemplo, provedores para validação de CEP, endereço de e-mail, nomes, sobrenomes e entre outros atributos que os usuários podem fornecer. Deste modo, logo após o cadastro o provedor de IdP direciona o usuário para acessar cada um dos provedores de validação e realizar o processo de abertura e disseminação do dado a ser validado para cada um dos serviços de validação de modo a concluir o seu cadastro.

O processo de validação ocorre de modo assíncrono, ou seja, há uma solicitação de validação, enviado do IdP com a participação do usuário para o Provedor de Validação (PV) e posteriormente a consulta sobre o resultado da validação por parte do IdP ao PV sem a participação do usuário e armazenamento desse resultado.

Os passos realizados para executar a validação dos PIIs cifrados são apresentados com o DFD da Figura 18, os processos em amarelo são executados no IdP, em verde é representado o processo de disseminação que será detalhado mais adiante e o conjunto do processos em laranja é a execução do serviço de validação pelo PV.

Os processos apresentados no DFD da Figura 18 são detalhados a seguir, esses processos são iniciados na etapa final do cadastro apresentado no diagrama da Figura 17.

- O IdP, após receber os atributos do usuário inicia o procedimento de validação dos dados recebidos;
- Para cada PII que for necessário uma conferência quanto a sua veracidade, deve-se executar uma chamada no PV responsável por sua validação;
- Deve ser executado o processo de disseminação de PII, descrito na Figura 19;

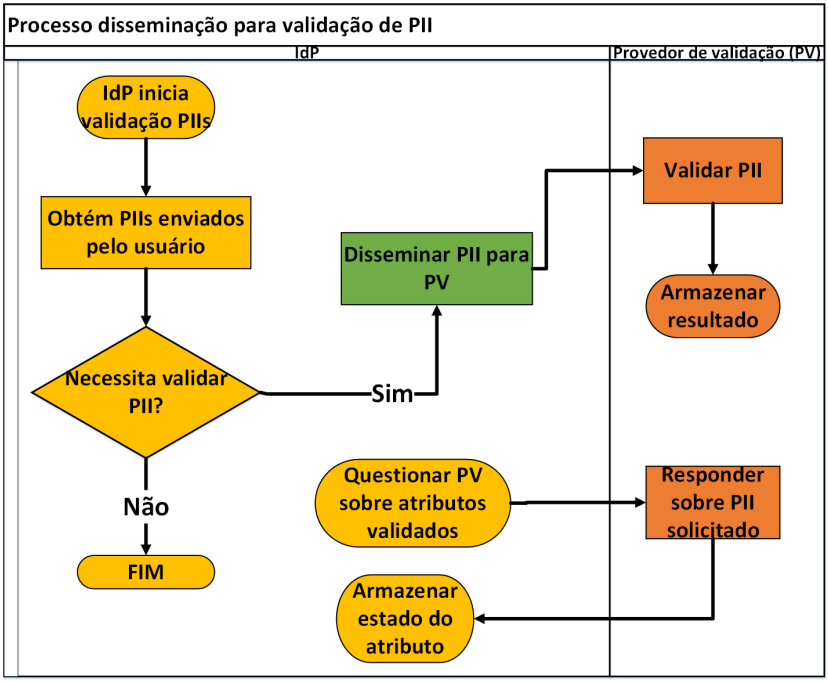


Figura 18 – Validação de PII cifrado.

- Após a disseminação do PII para o PV, o mesmo pode realizar o processo de conferência e comprovação da veracidade do PII;
- O resultado da validação é armazenado no PV para futura conferência do IdP;
- De modo assíncrono, o IdP consulta o PV sobre a validação de determinado atributo, isso é possível devido ao fato de que no momento da disseminação para validação o IdP envia junto ao PV um identificador da disseminação;
- De posse do resultado o IdP armazena-o e pode tomar as medidas cabíveis, caso o PII informado pelo usuário não seja confiável e/ou válido.

Em decorrência do processo de criptografia aplicado nos atributos dos usuários que são armazenados, o processo de disseminação entre

IdPs e SPs teve que ser alterado, caso contrário os atributos seriam enviados ao SP criptografados, o que inviabilizaria sua utilização.

A abertura dos atributos, assim como foi o processo de criptografia, é realizada no lado do cliente, sem que ocorra o envio de chaves para o servidor, podendo ser realizada dos seguintes modos:

- Atributos cifrados com chave pública – nesse caso quando abertura é necessária, o usuário deverá entrar com sua chave privada do seu certificado para que então o processo de decodificação possa ser realizado;

- PIIs cifrados com palavra chave – caso o usuário tenha optado por cifrar os dados com uma chave derivada a partir de uma palavra chave, é solicitado que ele entre a palavra chave utilizada no momento do cadastro para que o par de chaves seja gerado, de onde obtém-se a segunda chave do par e realiza-se o processo de abertura.

O processo de abertura dos dados ocorre após a realização da autenticação, durante a fase de disseminação, quando os usuários teriam a possibilidade de deliberar sobre a abertura dos atributos a serem enviados ao SP. A Figura 19 apresenta o fluxograma das tarefas executadas para abrir os PIIs. Os processos em verde são realizados pelo usuário, os processos em azul representam a execução de uma biblioteca no lado do cliente, os processos em amarelo são redirecionamentos que a aplicação faz para o navegador do usuário, de modo a guiá-lo a uma determinada interface/função fora do seu escopo, os processos em vermelho no lado do SP representam o módulo de controle de segurança sendo executado e por fim os processos em laranja representam o entrega de um recurso/funcionalidade da aplicação para o usuário.

O diagrama da Figura 19 pode ser descrito em mais detalhes:

- Usuário apresenta as suas credenciais (nome de usuário e senha) que são checados no repositório de dados. Caso o processo de confirmação de identidade ocorra com sucesso inicia-se o fluxo de disseminação;

- IdP direciona o usuário para a interface em que ele deve deliberar sobre o acesso aos seus dados que será realizado pelo SP;

- Usuário seleciona os atributos que deseja disseminar dentro dos escopos solicitados pelo SP;

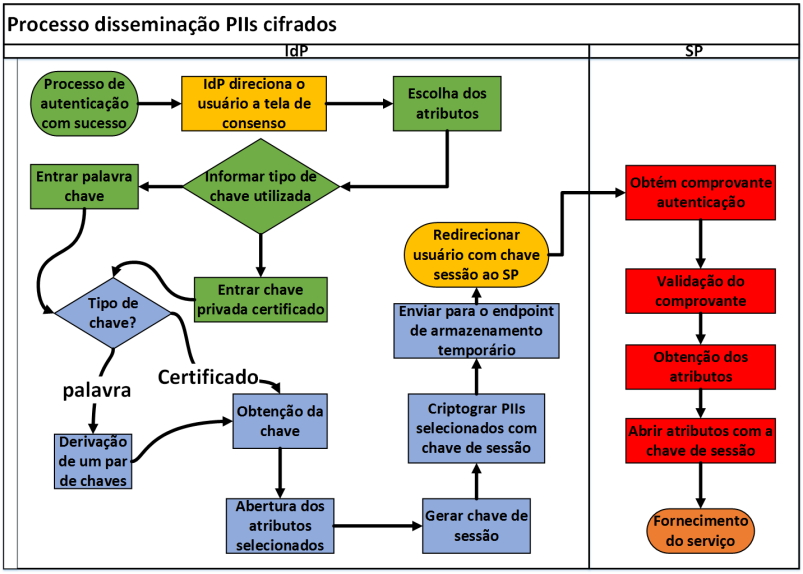


Figura 19 – Processo de disseminação alterado.

- Usuário é questionado sobre qual tipo de chave ele utilizou para realizar o processo de criptografia, o usuário entra com a chave ou palavra chave necessária para realizar o processo de abertura;
- Logo após a entrada da chave, o conjunto passos do processo de abertura é iniciado no lado do cliente;
- Obtém-se a chave necessária para realizar a abertura. Caso o usuário selecione palavra chave, deriva-se o par de chaves e utiliza-se a segunda chave do par. Caso contrário, a segunda chave do par de chaves do certificado que o usuário possui deve ser informada;
- O processo de abertura é realizado, somente nos PIIs que foram selecionados pelo usuário;
- É gerada uma chave de sessão para criptografar os PIIs abertos, de modo que quando eles sejam enviados ao armazenamento temporário, o IdP não possa se apropriar indevidamente deles;
- Os PIIs abertos são criptografados com a chave de sessão;
- Os PIIs são enviados ao ponto de armazenamento temporário (em memória), onde são armazenados, aguardando o processo de

solicitação dos atributos pelo SP. O tempo de vida desses atributos não ultrapassa o tempo limite para validação do *token* de autenticação gerado pelo IdP e enviado ao SP. Recomenda-se a utilização de um tempo de vida curto para a validação desse *token*, somente o necessário para o IdP realizar o processo de criação do *token* , envio ao SP e requisição de validação;

- Após o envio dos dados que serão armazenados temporariamente no IdP, o usuário é redirecionado ao SP com um comprovante de autenticação (*token*) e a chave de sessão gerada no navegador do usuário;
- SP obtém o comprovante de autenticação;
- São realizadas as validações necessárias, verificação se não é um ataque de repetição de requisição, se o comprovante está gerado no padrão correto, se ele não está expirado e entre outras;
- Após as validações do *token*, o SP solicita ao IdP os atributos liberados pelo usuário, após a resposta o IdP remove estes atributos da memória;
- SP abre os atributos com a chave de sessão recebida;
- Por fim o serviço/recurso solicitado pelo usuário é fornecido.

4.5.2 Adicionando suporte ao usuário no processo de disseminação

Esta subseção destina-se a apresentar a proposta com as medidas necessárias para tratar a ameaça 3, descrita com a Figura 15. Desta forma, foram utilizadas de forma combinada as propostas de uso de reputação das entidades federadas proposta em (SÁNCHEZ et al., 2012) e(BETGÉ-BREZETZ et al., 2012) com as propostas de uso de políticas para controlar o uso de dados apresentadas em (CHADWICK; FATEMA, 2012) e (BETGÉ-BREZETZ et al., 2013).

Birrell e Schneider (2013) argumentam que o controle da disseminação de PIIs é inconveniente por forçar os usuários a decidirem recorrentemente quais dados podem ser enviados a quais SPs. Em (ZHANG et al., 2007) é apresentado que para os usuários é difícil estabelecer com êxito suas políticas de disseminação de PII. Hansen, Schwartz e Cooper (2008) complementam que uma única configuração padrão para liberação de atributos não atenderia adequadamente as diferentes

necessidades dos usuários. Deste modo, propõe-se que os provedores de identidades utilizem a reputação dos provedores de serviços para auxiliar os usuários na decisão dos conjuntos de dados a serem disseminados.

Este trabalho não propõe um método para mensurar a reputação das entidades de uma federação, assume-se que existe um quantificador de reputação, como apresentado na Figura 20, este quantificador pode por exemplo ser uma implementação dos trabalhos (SÁNCHEZ et al., 2012; BETGÉ-BREZETZ et al., 2012), onde são apresentados modelos e métodos para quantificar a reputação dos membros de uma federação.

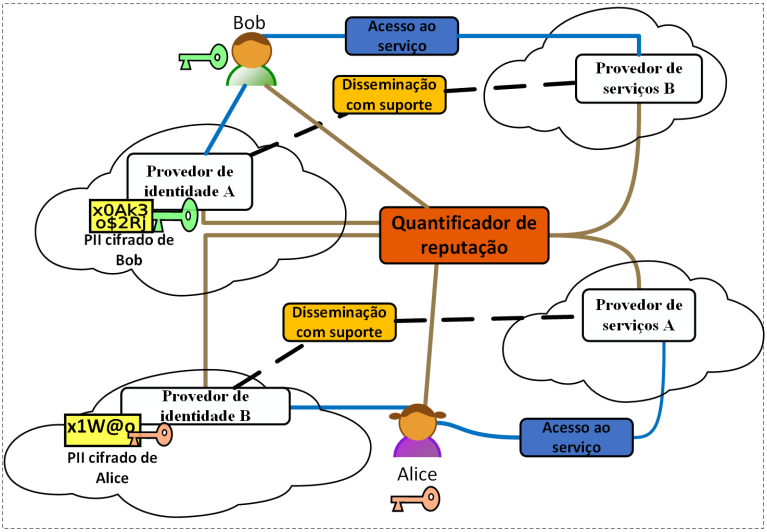


Figura 20 – Proposta para tratamento da ameaça 3.

O protocolo utilizado por este trabalho para a criação do ambiente federado (OpenID Connect) possui o conceito de *Scope*, que representa um conjunto de dados a ser enviado ao SP, sendo que o SP pode solicitar mais de um conjunto de dados ao mesmo tempo, e o cliente delibera sobre o envio ou não desses conjuntos. Assim, foi adicionado um atributo que representa o nível de reputação necessária para liberação daquele conjunto de dados ao SP no objeto *Scope*.

Com o uso de uma métrica de reputação no *scope* o IdP pode apresentar a interface de disseminação com *scopes* previamente selecionados ou não para facilitar o trabalho do usuário. O IdP utiliza políticas para indicar quais são os níveis de reputação necessário para

se acessar cada conjunto de atributos, auxiliando o usuário na decisão de quais *scopes* devem ser enviados ao SP.

A especificação do objeto *scope* (IETF, 2012) limita-se a definir que o objeto deve conter uma lista de chave e valor de elementos que compõem o escopo e que os mesmos devem ser documentados pelo IdP. Para tal este trabalho propõe uma representação para o objeto *scope* com os seguintes elementos:

- Nome (*name*) – um conjunto de literais que representa de forma curta o escopo. Atributo utilizado para apresentação ao usuário no IdP, representado na linha 2 da Figura 21;
- Identificador (*scope_id*) – um conjunto de literais que representa de forma curta e única um escopo, utilizado para solicitação dos escopos entre SP-IdP, representado na linha 3 da Figura 21;
- Reputação sugerida (*required_reputation*) – pontuação de reputação sugerida para o SP possuir para poder realizar o acesso aos atributos contidos no respectivo escopo, representado na linha 4 da Figura 21;
- Lista de atributos (*atributes*) – nomes de atributos que representam os PIIs do usuário. Esses valores são representados em literais que devem ser sensíveis a letras maiúsculas e minúsculas (*case sensitive*), representado na linha 5 da Figura 21.

A Figura 21 ilustra uma representação do elemento *scope* em formato JSON.

```
1   {
2   "name":"Perfil básico do usuário",
3   "scope_id":"perfil_basico",
4   "required_reputation":60,
5   "atributes":[
6           "nome",
7           "sobrenome",
8           "email"
9           ]
10  }
```

Figura 21 – Objeto escopo proposto em formato JSON.

O IdP não deve proibir que o usuário envie um *scope* mais completo ao SP, mesmo que ele considere isso um risco a privacidade do

usuário. Porém, ele deverá alertar sobre a reputação sugerida para que determinado conjunto de dados seja enviado.

O diagrama apresentado na Figura 22 representa os passos realizados para que o IdP possa prestar suporte e guiar os usuários no processo de disseminação. Os processos em amarelo são realizados no servidor IdP, a tarefa em verde é representada pelo usuário e os processos em azul são executados no navegador do usuário por uma biblioteca. Os processos aqui apresentados ocorrem durante a execução do redirecionamento do usuário para interface de consenso e disseminação, primeiro processo em amarelo representado na Figura 19.

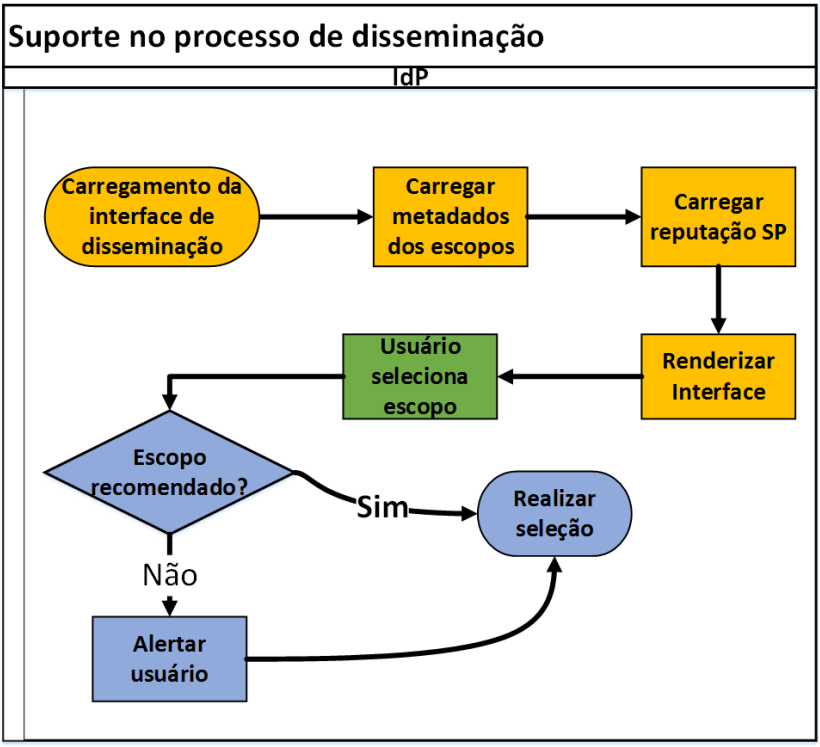


Figura 22 – Processo de suporte ao usuário durante disseminação.

Descrição da sequência de passos apresentados no digrama da Figura 22:

- Após o processo de autenticação quando o processo de criação da interface de disseminação é iniciado;

- São carregados os metadados dos escopos no formato descrito anteriormente;
- É carregada a reputação do SP com uma chamada a um *webservice* que atua em favor do quantificador de reputação;
- Inicia-se o processo de geração e envio da interface ao navegador do usuário;
- Usuário escolhe os escopos para disseminação;
- Biblioteca no navegador do usuário verifica se o escopo selecionado está dentro das políticas de recomendação do IdP;
- Caso não esteja dentro das políticas de segurança estabelecidas, o usuário é alertado sobre o risco envolvido na disseminação daquele escopo;
- *Scope* é selecionado.

Em um esforço inicial, este trabalho gerou um artigo em que é proposto o uso de políticas que são inseridas pelos usuários nos IdPs para auxiliar e automatizar o processo de disseminação (WEINGÄRTNER; WESTPHALL, 2014). Porém, o desenvolvimento e manutenção de políticas de liberação de dados como PIIs é uma tarefa complexa e árdua de ser realizada pelos usuários. Assim, este trabalho propõe que ao invés de repassar essa responsabilidade ao usuário, o provedor de identidade é que iria estabelecer as políticas de disseminação e aplicá-las de modo a auxiliar os usuários no processo de disseminação. Deste modo, o uso é transparente para os usuários, e ele recebe indicações sobre suas escolhas na interface de disseminação.

O nível de reputação do provedor de serviço não garante que ele não esteja sujeito a falhas de segurança e quebra de privacidade. Porém, um nível alto de reputação é um indicador de que a princípio o provedor está realizando o seu trabalho da forma correta, aplicando as políticas necessárias para garantir a segurança e privacidade dos atributos dos usuários.

5 DESENVOLVIMENTO DO MODELO DE PRIVACIDADE EM SISTEMAS FEDERADOS

Este capítulo destina-se a relatar as alterações que foram realizadas na ferramenta MITREID (MIT, 2014) que implementa o protocolo utilizado estendido com as propostas apresentadas na seção 4.5.

5.1 FERRAMENTAS

A Figura 23 ilustra as tecnologias utilizadas para o desenvolvimento das propostas aqui apresentadas. No ambiente de nuvem do laboratório foi utilizado o sistema operacional Debian 7.0 (DEBIAN, 2014) como base, em conjunto com os pacotes Xen hypervisor (XENPROJECT, 2015a) para virtualização e XAPI (XENPROJECT, 2015b) para criação de grupos de servidores e proporcionar o desenvolvimento do ambiente de nuvem. Para gerenciar a infraestrutura foi utilizado o Cloudstack versão 4.3.0 (APACHE, 2015a) como ferramenta de orquestração.

Na Figura 23 são apresentadas as tecnologias utilizadas tanto no lado do IdP quanto do SP. Tanto no SP quanto no IdP utilizou-se o *framework* Spring, com as tecnologias Java e Javascript com o framework jQuery. Ainda no IdP foi utilizado o container de aplicação Tomcat e a implementação do protocolo OpenId Connect do Massachusetts Institute of Technology (MIT) (MIT, 2014).

5.2 AJUSTES PRELIMINARES NO *FRAMEWORK*

Quando foram iniciados os trabalhos para desenvolver as propostas com o *framework* MITREID(MIT, 2014), fora encontrados problemas no que tange a compatibilidade da biblioteca com a especificação e a nível de arquitetura do software. Assim, esta seção destina-se a descrever essas alterações de forma sucinta.

5.2.1 Melhorar a compatibilidade da biblioteca desenvolvida do SP com sua especificação

No início dos trabalhos não havia um IdP sendo executado no LRG utilizando a especificação OpenID Connect. Deste modo, foi utilizado um IdP de terceiros que implementasse a especificação OpenId

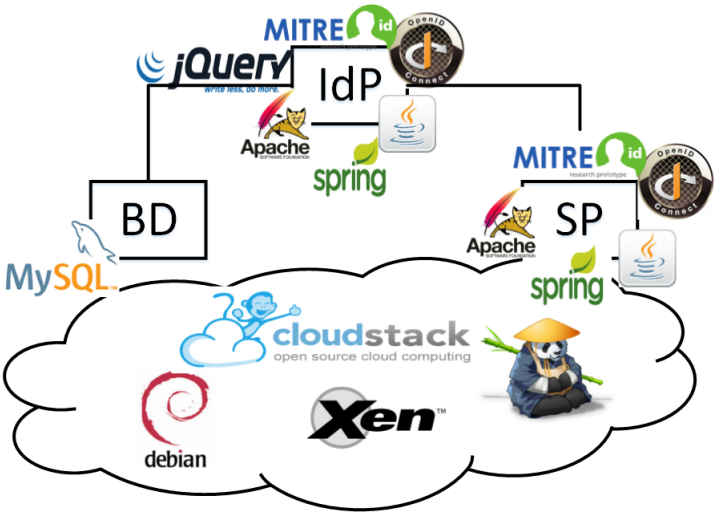


Figura 23 – Tecnologias utilizadas

Connect, como por exemplo o Google Accounts (GOOGLE, 2014). Pensou-se que por estar sendo utilizado um protocolo padronizado, não deveriam haver problemas de integração, pelo fato de as mensagens e o fluxo de interações serem definidos em um protocolo compartilhado entre as partes.

No decorrer dos testes, notou-se que durante a realização da requisição para validação do comprovante de autenticação (*token*), a especificação definia que poderia ser utilizado um atributo chamado "*nonce*", que seria um valor único para cada requisição, de modo a facilitar a identificação de ataques. Porém, segundo a especificação o uso desse atributo deve ser opcional, sendo assim as bibliotecas do SP deveriam permitir a ativação e/ou desativação desse atributo para um IdP específico. No caso do IdP Google Accounts (GOOGLE, 2014), esse atributo não poderia ser utilizado, pois não era suportado pelo mesmo.

Deste modo, foi realizado um ajuste na biblioteca para torná-la compatível com a especificação, possibilitando a ativação/desativação do uso do atributo "*nonce*". As alterações realizadas são descritas a seguir.

No componente conhecido como "*openid-connect-common*", classe *org.mitre.openid.connect.config.ServerConfiguration* foi adicionado um atributo booleano chamado "useNonce", que é inicializado com valor *true* para manter compatibilidade, os respectivos métodos "*get*"e

"*set*"foram criados. Essa classe é utilizada para representar as configurações de um servidor IdP.

No componente conhecido como "*openid-connect-client*", interface *org.mitre.openid.connect.client.service.AuthRequestUrlBuilder*, existe um método chamado "*buildAuthRequestUrl*"que é responsável por criar a URL de solicitação de autenticação, para a qual o usuário é direcionado. Um dos parâmetros que o método "*buildAuthRequestUrl*"recebe é o valor "*nonce*". Conforme ilustrado na Figura 24 foram criados duas assinaturas de método para o "*buildAuthRequestUrl*", uma que recebe um valor "*nonce*"e outra sem.

```
public interface AuthRequestUrlBuilder {
    /**
    * @param serverConfig
    * @param clientConfig
    * @param redirectUri
    * @param nonce
    * @param state
    */
    public String buildAuthRequestUrl(
        ServerConfiguration serverConfig,
        RegisteredClient clientConfig, String
        redirectUri, String nonce, String state,
        Map<String, String> options);

    /**
    * @param serverConfig
    * @param clientConfig
    * @param redirectUri
    * @param state
    */
    public String buildAuthRequestUrl(
        ServerConfiguration serverConfig,
        RegisteredClient clientConfig, String
        redirectUri, String state, Map<String,
        String> options);
}
```

Figura 24 – Interface AuthRequestUrlBuilder alterada.

Após realizar a alteração na interface, foi necessário codificar o

método nas classes que implementam a interface alterada, de modo que elas passam a ter dois métodos: um que utiliza "*nonce*"e outro não. As seguintes classes foram alteradas, encontrando-se no pacote *org.mitre.openid.connect.client.service.impl*.

- *PlainAuthRequestUrlBuilder*;
- *EncryptedAuthRequestUrlBuilder*;
- *SignedAuthRequestUrlBuilder*.

Depois de realizar as alterações nas classes que geram a URL de solicitação de autenticação no SP, foi necessário alterar a classe que utiliza essas implementações. Deste modo, dependendo das configurações do servidor IdP será utilizado o método que necessita ou não do atributo "*nonce*". A classe que faz esse procedimento é chamada de *org.mitre.openid.connect.client.OIDCAuthenticationFilter*, o método alterado se chama "*getAuthRequestUrl*". A alteração consistiu de um teste condicional, conforme ilustrado na Figura 25, linha de número 10.

Com essa alteração, a aplicação de testes desenvolvida pode realizar a solicitação de autenticação de um usuário para o IdP Google Accounts (GOOGLE, 2014), de modo que o usuário poderia se autenticar utilizando uma conta do Google.

5.2.2 Tornar a biblioteca do IdP extensível

O conjunto de bibliotecas que possibilitam a criação de um servidor IdP desenvolvido pelo projeto MITREID utiliza o conceito *overlay* da ferramenta Maven (APACHE, 2015b), que é utilizada para fazer a gerência de dependências de projetos. Com essa tecnologia é possível, por exemplo, criar uma aplicação base, com todos os arquivos necessários e um funcionamento inicial de exemplo, que pode ser estendida de modo fácil e prático sem duplicação de arquivos, com a simples sobrescrita de arquivos.

A aplicação base para utilizar o método de *overlay* do Maven desenvolvida pelo MIT chama-se "*openid-connect-server-webapp*". Foi criado um projeto Maven no Eclipse (ECLIPSE, 2015) que utiliza essa aplicação como base para desenvolver um IdP que utilizasse uma base de dados de testes que está na nuvem do LRG. Porém, durante o desenvolvimento foram encontrados alguns problemas estruturais para realizar a extensão da aplicação.

```
...
Muito código aqui
...
protected String getAuthRequestUrl(
ServerConfiguration serverConfig,
RegisteredClient clientConfig,
String redirectUri, String state,
Map<String, String> options,
HttpSession session) {
if(serverConfig.isUseNonce()){
String nonce = createNonce(session);
return authRequestBuilder.buildAuthRequestUrl(
    serverConfig, clientConfig, redirectUri,
    nonce, state, options);
}
return authRequestBuilder.buildAuthRequestUrl(
    serverConfig, clientConfig, redirectUri,
    state, options);
}
...
Muito mais código aqui
...
```

Figura 25 – Classe OIDCAuthenticationFilter alterada.

Na aplicação base as configurações para instanciar os objetos que lidam com a obtenção dos atributos dos usuários do repositório de dados estavam sendo criados com anotações Java, o que impossibilitava a alteração das configurações facilmente, sem a alteração e compilação da aplicação base novamente.

Assim, foram removidas as anotações "*@Autowired*"dos atributos e as anotações "*@Service*"das seguintes classes que encontram-se no pacote *org.mitre.openid.connect.service.impl*.

- *DefaultUserInfoService*

- *DefaultUserInfoUserDetailsService*

Também foi removida a anotação "*@Component*"da classe *JpaUserInfoRepository* do pacote *org.mitre.openid.connect.repository.impl*. Esse objeto somente deve ser criado na aplicação base do *overlay*, sendo necessária a sua implementação e alteração em uma aplicação real. Assim, com a remoção da anotação ele deixa de ser criado quando não é utilizado.

Com a remoção das anotações das classes mencionadas acima, foi necessário repassar as suas configurações para instanciar os objetos para o arquivo XML conhecido como "*user-context.xml*", que é o arquivo que concentra as informações de configuração dos objetos que controlam o acesso ao repositório de dados. A Figura 26 demonstra como ficaram as configurações no XML das classes que tiveram suas anotações removidas.

Além das alterações descritas acima foi realizada a padronização da interface *org.mitre.openid.connect.model.UserInfo*. Essa interface deve ser utilizada pelas implementações do IdP para representar um objeto de negócio que armazenaria os atributos dos usuários. Esse tipo de interface deve possuir somente os métodos essenciais para seu funcionamento dentro do *framework*.

Porém, por uma falha de projeto, tentou-se prever todos os dados possíveis que um objeto que represente o usuário viesse a ter. Com isso a interface tinha métodos que não eram necessários. A Figura 27 demonstra como ficou a interface proposta, utilizando somente os métodos essenciais para o desenvolvimento e utilização do *framework*.

Somente depois de realizar esses ajustes é que foi possível começar o trabalho de desenvolvimento das propostas apresentadas no capítulo anterior.

```
...
Muito mais configuração em XML aqui
...
    <!-- Beans that manage user informations -->
    <bean id="userInfoUserDetailsService" class=
        "org.mitre.openid.connect.service.impl.
        DefaultUserInfoUserDetailsService">
         <property name="repository" ref="
            jpaUserInfoRepository" />
    </bean>
    <bean id="defaultUserInfoService" class="org
        .mitre.openid.connect.service.impl.
        DefaultUserInfoService">
         <property name="userInfoRepository" ref=
            "jpaUserInfoRepository" />
         <property name="clientService" ref="
            defaultOAuth2ClientDetailsEntityService
            " />
         <property name="
            pairwiseIdentifierService" ref="
            uuidPairwiseIdentiferService" />
    </bean>
    <bean id="jpaUserInfoRepository" class="org.
        mitre.openid.connect.repository.impl.
        JpaUserInfoRepository" />

<!-- ScopeTranslator -->
    <bean id="scopeClaimTranslator" class="org.
        mitre.openid.connect.service.impl.
        DefaultScopeClaimTranslationService"/>
</beans>
```

Figura 26 – Criação dos objetos em user-context.xml

```
1
public interface UserInfo {
3       /**
        * @return the userId
5       */
        public abstract String getSub();
7
        /**
9       * @param sub the userId to set
        */
11      public abstract void setSub(String sub);

13
        /**
15      * @return the name
        */
17      public String getName();

19      /**
        * Serialize this UserInfo object to JSON
21      *
        * @return
23      */
        public abstract JsonObject toJson();
25
        /**
27      * receives a JsonObject and populate this
           object with those attributes.
        *
29      */
        public void fromJson(JsonObject obj);
31  }
```

Figura 27 – Interface UserInfo ajustada.

5.3 CADASTRO DE USUÁRIOS NO IDP

Para criação de formulários que atendessem a proposta de armazenamento de PII cifrado pelo usuário, foi desenvolvida uma biblioteca de *tags* utilizando tecnologias *Java web* que servem de estrutura básica para a criação dos formulários. Os dados que são informados pelos usuários são cifrados no navegador com uma chave antes de serem enviados. As chaves do usuário não são enviadas ao servidor em nenhum momento.

Em conjunto com as *tags*, foi utilizado a linguagem de programação *Javascript* em conjunto com a biblioteca de criptografia assimétrica Cryptico (TERRELL, 2012). A biblioteca Cryptico utiliza métodos de criptografia assimétrica no lado do cliente, sendo construída sobre a biblioteca de criptografia desenvolvida em Stanford (WU, 2009). Para gerar as chaves de criptografia de acordo com o padrão Password-Based Key Derivation Function 2 (PBKDF2) (KALISKI, 2000) utilizou-se a *Stanford Javascript Crypto Library (SJCL)* (STARK; HAMBURG; BONEH, 2009) (STARK; HAMBURG; BONEH, 2015).

A Figura 28 apresenta a *tag* utilizada para entrada dos atributos. Enquanto que a Figura 29 demonstra um trecho do código da *tag* no formulário de exemplo criado, está sendo apresentado o bloco de código que é responsável por obter os dados informados pelo usuário e realizar o processo de criptografia.

A Figura 30 demonstra a utilização dos componentes criados que proporcionam a criação de uma tela de cadastro que possibilita a codificação dos dados antes do envio ao provedor de identidades. Destaca-se que os campos de login e senha não são cifrados com a chave do usuário de modo a permitir a execução do processo de autenticação. O armazenamento da senha é feito com a utilização de um algoritmo de *hash*, para que nenhum administrador descubra a senha do usuário facilmente.

O processo de disseminação de dados entre SPs e IdPs teve que ser estendido em decorrência do uso de PIIs cifrados, conforme demonstrado na Figura 31, caso contrário os atributos seriam enviados ao SP cifrados.

Para realizar as alterações no processo de disseminação proposto, foi necessário criar um conjunto de classes que fazem o armazenamento do dado temporariamente. Foi criada uma classe chamada *br.ufsc.lrg.openid.connect.controller.UserInfoTempController* que faz o recebimento da requisição AJAX que envia os dados do usuário para armazenamento temporário. A Figura 32 ilustra como ficou essa classe.

```
<%@ tag language="java" pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<%@ attribute name="label" required="true" %>
<%@ attribute name="name" required="true" %>
<%@ attribute name="labelWidth" required="false" %>
<%@ attribute name="type" required="false" description="default value is 'text'" %>
<%@ attribute name="shouldEncrypt" required="false" description="default value is true" %>

<c:if test="${empty shouldEncrypt}">
    <c:set var="shouldEncrypt" value="true" />
</c:if>
<c:if test="${empty type }">
    <c:set var="type" value="text" />
</c:if>
<c:if test="${empty labelWidth}">
    <c:set var="labelWidth" value="150" />
</c:if>

<div style="width: 100%; display:inline-flex;">
    <div style="width: ${labelWidth}px;" >
        <b>${label}:</b>
    </div>
    <div style="display: inline;">
        <input type="${type}" class="input-xlarge ${ shouldEncrypt ? 'inputPrivacy' : ''}"
        name="${name}" id="${name}" />
    </div>
</div>
```

Figura 28 – Tag utilizada para entrada dos atributos (*input*).

```
var cifrarDadosPalavraChave = function (chave){
    var Bits = 1024;
    var parDeChaves = cryptico.generateRSAKey(chave, Bits);
    var chavePublica = cryptico.publicKeyString(parDeChaves);

    cifrarDados(chavePublica);
}
var cifrarDados = function (chave){
    jQuery('.inputPrivacy').each(
            function (index){
                cifrarDadoInput(jQuery(this), chave);
                });
};
var cifrarDadoInput = function (input, chave){
    var dadoASerCifrado = input.val();

    var dadoCifrado = cryptico.encrypt(dadoASerCifrado , chave);
    var stringDadoCifrado = dadoCifrado.cipher

    input.val(stringDadoCifrado);
};
```

Figura 29 – Tag formulário (*form*) para entrada dos atributos.

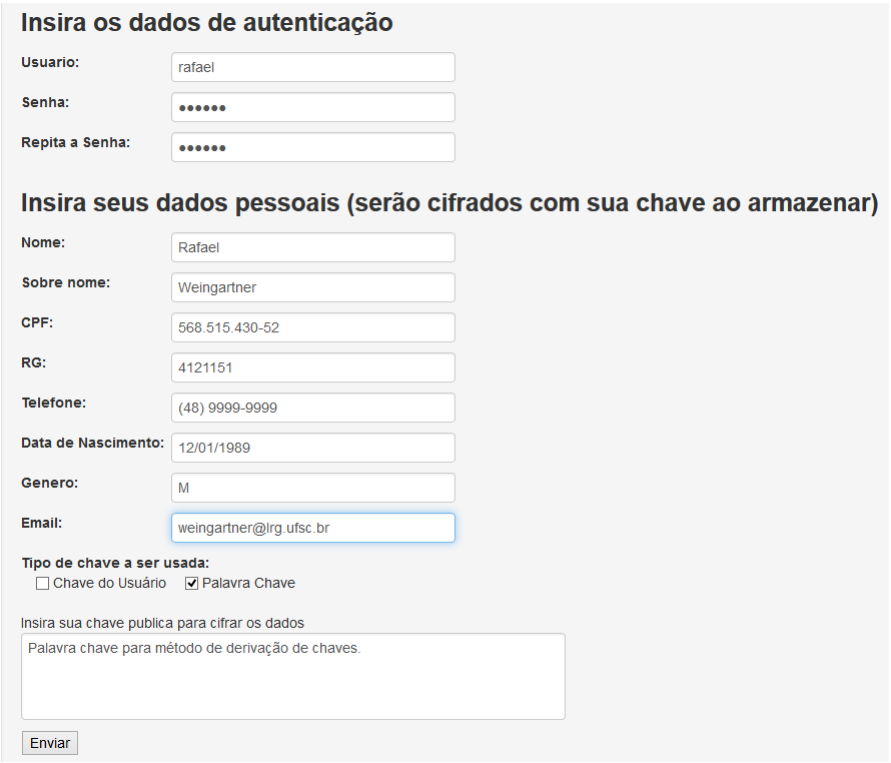


Figura 30 – Cadastro dos dados com privacidade.

Figura 31 – Dados cifrados no IdP.

```
@Controller
@RequestMapping(method=RequestMethod.POST, value
    ="/usuarioInfo")
public class UserInfoTempController {

    @Autowired
    private LrgUserInfoTempService
        lrgUserInfoTempService;

    @RequestMapping("/adicionarAtributos")
    public void
        armazenarTemporariamenteAtributosUsuario(
        @RequestBody LrgUserInfo lrgUserInfo){
    LrgTempUserInfoHolder lrgTempUserInfoHolder
        = new LrgTempUserInfoHolder(lrgUserInfo);
    this.lrgUserInfoTempService.save(
        lrgTempUserInfoHolder);
    }
}
```

Figura 32 – Endpoint para armazenamento temporário dos atributos.

```
...
Muito mais configuração em XML aqui
...
<bean id="userInfoUserDetailsService" class="org
    .mitre.openid.connect.service.impl.
    DefaultUserInfoUserDetailsService">
     <property name="repository" ref="
        lrgUserInfoTempService" />
</bean>
...
Muito mais configuração em XML aqui
...
```

Figura 33 – Injeção do objeto *lrgUserInfoTempService* no *userInfoUserDetailsService*

Adicionalmente, criou-se a classe *LrgUserInfoTempService* no pacote *br.ufsc.lrg.openid.connect.service* que é injetada no objeto *userInfoTempController* e faz o armazenamento dos dados temporariamente. Essa classe de serviço também é injetada na classe responsável por atender a validação da confirmação de autenticação que é realizado pelo SP (o nome do objeto é *userInfoUserDetailsService*). A Figura 33 demonstra onde esse objeto de serviço foi injetado, enquanto que a Figura 34 demonstra a parte mais importante dessa implementação, que é o armazenamento e obtenção dos dados a partir do login do usuário ou do identificador conhecido como *sub*, que é utilizado no OpenId Connect (O valor de *sub* no caso é um identificador único do usuário).

Foi também necessário alterar um arquivo conhecido como "*approve.jsp*", esse arquivo é responsável por gerar o HTML da página de disseminação. Foi adicionada uma função Javascript para enviar o dado para o serviço de armazenamento temporário, conforme ilustrado na Figura 35. A função de abertura do dado também teve que ser incluída nesse arquivo, a função desenvolvida é apresentada na Figura 36.

```
@Service
public class LrgUserInfoTempService
    implements UserInfoRepository{

private Map<String, UserInfo>
    mapUserBySubUserTempAttributes = new
    HashMap<String, UserInfo>();
private Map<String, UserInfo>
    mapUserByLoginUserTempAttributes = new
    HashMap<String, UserInfo>();
private long
    valorMaximoParaArmazenamentoTemporario =
    10000;

...
códigos
...
@Override
public UserInfo getBySubject(String sub) {
LrgTempUserInfoHolder userInfo = (
    LrgTempUserInfoHolder) this.
    mapUserBySubUserTempAttributes.get(sub);
userInfo.setUsed(true);
return userInfo;
}
@Override
public UserInfo getByUsername(String
    username) {
LrgTempUserInfoHolder lrgTempUserInfoHolder
    = (LrgTempUserInfoHolder) this.
    mapUserByLoginUserTempAttributes.get(
    username);
lrgTempUserInfoHolder.setUsed(true);
return lrgTempUserInfoHolder;
}
```

Figura 34 – Parte da implementação do objeto de armazenamento de atributos temporários.

```
var enviarAtributosParaEndpointTemporario =
    function () {
var atributosUsuario = {};

jQuery('input.inputDisseminacao').each(function(
    index){
     atributosUsuario[''+jQuery(this).attr('name'
         )]= ''+jQuery(this).val();;
});
jQuery.ajax({
     url: '${pageContext.request.contextPath}/
         usuarioInfo/adicionarAtributos',
     type: 'post',
     dataType: 'json',
     data: atributosUsuario
     });
}
jQuery('input.btn-success').click(
    enviarAtributosParaEndpointTemporario);
```

Figura 35 – Javascript com função AJAX para envio dos dados para o serviço de armazenamento temporário.

```
var decrypt = function(input) {
    if ($('#usedkey').attr('name') == 'userkey')
        {
        var chave = $('#usedkey').val();
        var StringDecifrada = cryptico.decrypt(
            input.val(), chave);
    } else {
        var PassPhrase = $('#usedkey').val();
        var Bits = 1024;
        var chave = cryptico.generateRSAKey(
            PassPhrase, Bits);
        var chavePublica = cryptico.
            publicKeyString(chave);

        var StringDecifrada = cryptico.decrypt(
            input.val(), chavePublica);
    }
    substituiInput(StringDecifrada);
};
```

Figura 36 – Javascript função abertura atributo.

5.4 ADIÇÃO DE SUPORTE AO USUÁRIO NO PROCESSO DE DISSEMINAÇÃO

Para a proposta de suporte ao usuário no processo de disseminação, com uso da reputação do provedor de serviços, é necessário obter a reputação do SP.

Bastaria tornar essa métrica disponível para o JSP responsável por gerar a página de disseminação e possibilitar a criação de funções para guiar os usuários durante o processo. Porém, o componente que o *framework* MITREID utiliza para fazer esse processo não permite uma extensão facilmente, pois, assim como ocorre no caso do IdP, foram utilizadas anotações nas classes da biblioteca, o que inviabiliza uma extensão de maneira fácil.

Assim, essa situação foi contornada com a criação de um filtro web, que intercepta as requisições que a aplicação IdP recebe, quando é detectada uma requisição para o *endpoint* da aplicação que trata o processo de disseminação, é realizado o processo de carregamento da reputação do SP que solicitou aquela autenticação. Esse valor é então repassado para o componente que gera a página de disseminação.

A especificação do protocolo define que na requisição de autenticação realizado pelo SP ao IdP, é enviado um atributo chamado *redirect_uri* que contém a URL de redirecionamento ao SP após o processo de disseminação. Assim, foi feito uso desse atributo para obter um identificador único do SP na federação, esse identificador consiste do endereço completo do SP. De posse desse identificador é chamado um serviço web que é responsável por quantificar a reputação e retornar ao IdP. A reputação é então adicionada no objeto *request* que vai ser disponibilizado ao JSP que renderiza a página de disseminação, o nome do atributo utilizado para o valor de reputação é *SP_REPUTATION*.

Com essa alteração é possível alterar o JSP para que, com base na reputação recebida, seja proporcionado suporte ao usuário sobre quais atributos e escopos são mais aconselhados a serem disseminados, visto que o escopo possui um nível de reputação recomendado, definido em política pelo IdP e armazenado no repositório de dados. A tela de disseminação com as alterações é ilustrada nas figuras 37 e 38

Com as alterações apresentadas os usuários passam a ter uma maior ciência sobre quais dados estão sendo disseminados e a obter suporte sobre quais atributos e escopos são mais recomendados a serem disseminados para cada provedor de serviços. Assim, pode-se reduzir os perigos envolvidos com a disseminação de determinados conjuntos de atributos ao mesmo tempo em que contempla-se a legislação, forne-

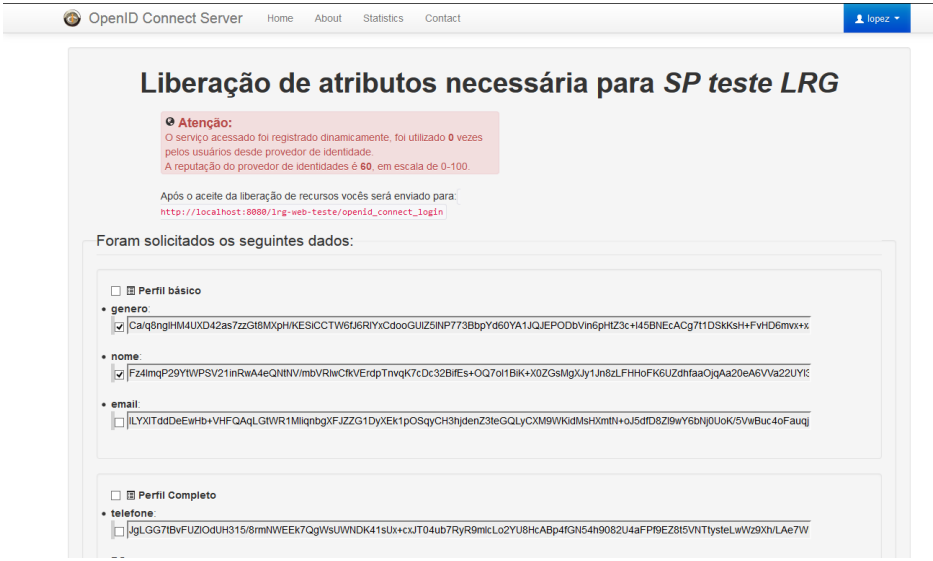


Figura 37 – Interface de disseminação aprimorada.

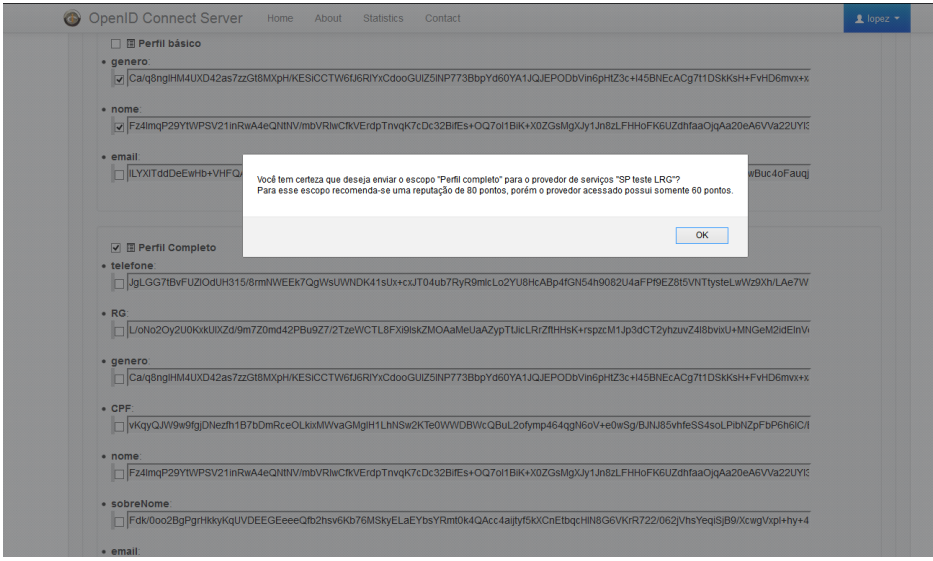


Figura 38 – Alerta para seleção de atributo/escopo não recomendado.

cendo informações aos usuários sobre quais dados estão sendo disseminados e dando poder para o usuário deliberar sobre essa disseminação.

5.5 VALIDAÇÃO DA PROPOSTA

Para validar as propostas e testar se as mesmas resolveram as ameaças modeladas e discutidas anteriormente, utilizou-se algumas das recomendações apresentadas em (SHOSTACK, 2014). Quando se modela a ameaça, pode-se que verificar com uso de um *checklist* se as medidas tomadas sanaram as ameaças modeladas.

Assim, das ameaças que foram modeladas, tem-se:

- Ameaça 1 – Falta de ciência na disseminação de atributos, apresentada com a Figura 13;
- Ameaça 2 – Carência no controle dos atributos dos usuários, ilustrada com a Figura 14;
- Ameaça 3 – Processo de disseminação complicado e sem possibilidade de customização, apresentada com a Figura 15.

Para tratar a ameaça 1 foi alterada a página de disseminação, de modo que os usuários vejam quais são os conjuntos de atributos que estão sendo solicitados pelo SP. Assim, com o modelo e implementações realizadas esse problema foi solucionado.

A ameaça 2 tratava sobre o controle dos PIIs armazenados nos IdPs, para proporcionar tal controle sobre o seu uso, foi proposto e implementado o uso de PIIs cifrados com uma chave do usuário. Somente o detentor da chave, consegue fazer uso destes atributos. Administradores e/ou desenvolvedores que acessam o repositório de dados somente tem acesso aos dados cifrados, inviabilizando o seu uso sem a abertura dos mesmos com o uso da chave que somente o usuário deve possuir. Assim, essa ameaça foi considerada solucionada com a aplicação das medidas propostas.

No que tange a ameaça 3, quanto a customização e suporte durante o processo de disseminação, foi proposta a criação de políticas para informar os usuários sobre quais conjuntos de PIIs são mais adequados a serem liberados, de acordo com a reputação do SP. Além disso, foi proposta e implementada a customização dos objetos escopos, proporcionando aos usuários meios para definir quais atributos de cada escopo podem ser disseminados para o SP. Com essas medidas a ameaça 3 pode ser considerada sanada, em decorrência de que os

usuários tem a liberdade de customização dos atributos disseminados em conjunto com o uso de mensagens informativas que são apresentadas para os usuários de acordo com políticas estabelecidas pelo IdP.

6 CONCLUSÃO

Os três questionamentos que motivaram e iniciaram essa pesquisa sobre os PIIs dos usuários de ambientes federados foram tratados com a apresentação da bibliografia no capítulo 2. Usuários são os donos dos dados armazenados em IdPs, por isso eles devem ter ciência de quando estes dados estão sendo utilizados e/ou disseminados para terceiros (ORAWIWATTANAKUL et al., 2010). Estes mecanismos deveriam existir para proteger os usuários contra disseminação e processamento de dados não autorizados.

Existem legislações e padrões para gerenciamento de dados que buscam proporcionar garantias para usuários sobre seus dados armazenados em provedores como IdPs e SPs. Com todo o arcabouçou legal e guias de melhores práticas, usuários deviam ter total controle dos seus PIIs armazenados em IdPs.

Porém, no que diz respeito às atuais ferramentas utilizadas para criar ambientes federados, após a apresentação do modelo corrente de gerenciamento de identidades e ferramentas, pode-se afirmar que atualmente não há métodos de controle efetivo para PIIs que podem ser utilizados pelos usuários, bem como quanto à ciência que somente algumas ferramentas aplicam medidas de modo a proporcioná-la.

Enquanto os trabalhos (SÁNCHEZ et al., 2012), (BETGÉ-BREZETZ et al., 2012), (BETGÉ-BREZETZ et al., 2013) e (CHADWICK; FATEMA, 2012) abordaram privacidade na nuvem respectivamente com uso de métricas de reputação e aplicando políticas de acesso para controlar os dados, Weingärtner e Westphall (2014) propuseram o armazenamento de atributos de identificação cifrados com uma chave do usuário nos IdPs e políticas de disseminação de dados que são inseridas pelos usuários nos IdPs para auxiliar no processo de disseminação.

Esta dissertação une as propostas apresentadas em (SÁNCHEZ et al., 2012), (BETGÉ-BREZETZ et al., 2012), (BETGÉ-BREZETZ et al., 2013), (CHADWICK; FATEMA, 2012) e (WEINGÄRTNER; WESTPHALL, 2014), utilizando políticas de liberação de dados que fazem uso da reputação dos provedores para auxiliar o processo de disseminação dos usuários e tentar reduzir os riscos envolvidos na liberação de certos conjuntos de atributos. O trabalho também armazena PIIs cifrados em IdPs, de modo a evitar que entidades curiosas/maliciosas nos IdPs violem a privacidade dos atributos de identificação dos usuários.

Deste modo, a proposta desta dissertação abrange os paradigmas de privacidade apresentados em (DIAZ; GÜRSES, 2012), abordando

privacidade sob o aspecto de controle, confidencialidade e social. A Tabela 1 compara os trabalhos relacionados e o conjunto de propostas para redução de riscos a privacidade dos usuários desta dissertação.

As propostas dessa dissertação abordam de um modo mais completo a privacidade dos usuários nos provedores de identidades. Com a proposta de cadastro de atributos de identificação cifrados nos IdPs, ilustrado com as Figuras 30 e 31 o usuário passa a ter uma camada de proteção sobre seus dados que são armazenados nos IdPs. Assim, dificulta-se uso de atributos de identificação sem a ciência e participação dos usuários.

Com a proposta de suporte no processo de disseminação, o usuário recebe um suporte de quais conjuntos de dados são aconselháveis a serem disseminados e caso ele marque um conjunto não aconselhado, ele será alertado sobre a liberação daquele conjunto, e que o mesmo pode vir a ferir sua privacidade.

Apesar dos benefícios obtidos, a proposta de PIIs cifrados pode tornar-se inconveniente, em decorrência de que a cada acesso que o usuário faz no provedor de serviço, força-se a realização do processo de disseminação recorrentemente, para abrir os dados que são enviados ao SP. Por um lado ganhou-se em controle, privacidade e segurança, por outro se perdeu em usabilidade, por forçar repetidamente o usuário a abrir o dado a ser enviado ao SP.

Apesar dos impactos gerados pela proposta em termos de usabilidade, os ganhos de controle e na privacidade dos atributos de identificação compensam a sua aplicação.

A utilização de métodos e mecanismos para tratar a privacidade em sistemas de gerenciamento de identidades é de grande importância, frente ao uso cada vez maior de sistemas federados e da expansão desses sistemas com o uso de computação em nuvem.

O tratamento da privacidade em sistemas de gerenciamento de identidades em ambientes dinâmicos como o caso de nuvens computacionais é um caminho ainda pouco explorado, não se tem uma resposta definitiva do que pode e deve ser tratado quanto à privacidade e que níveis e tipos de privacidade devem ser abordados nesses ambientes.

6.1 CONTRIBUIÇÕES

As principais contribuições alcançadas durante a pesquisa e desenvolvimento deste trabalho são:

- Identificação e discussão dos problemas quanto a privacidade nos

atributos de identificação dos usuários que são armazenados em provedores de identidades em sistemas federados;

- Proposta de um modelo para reduzir os riscos a privacidade dos atributos armazenados em IdPs;
- Proposta do uso de reputação para auxiliar e guiar os usuários no processo de disseminação.
- Definição de um método para facilitar a integração entre IdPs e SPs, de modo que os mesmos possam adaptar o seu funcionamento de acordo com o protocolo que cada um está utilizando;
- Aprimoramento da implementação do OpenId Connect (MITREID).

Esse trabalho resultou na publicação do artigo (WEINGÄRTNER; WESTPHALL, 2014), que apresenta os trabalhos iniciais realizados nessa dissertação. O artigo foi publicado em: *The Eighth International Conference on Emerging Security Information, Systems and Technologies – SECURWARE 2014*, classificado como B3, pelo Qualis CAPES.

6.2 TRABALHOS FUTUROS

Como trabalhos futuros pretende-se investigar meios de adicionar mecanismos para prover controle dos atributos dos usuários disseminados aos SPs. Um modo de controlá-los quando estes são recebidos no SP, por exemplo, indicando por quanto tempo aqueles dados podem ser armazenados, quem pode utilizar, que tipo de processamento e entre outras questões.

Formalizar e codificar um acordo de protocolo de modo a facilitar a integração das propostas em um ambiente funcional. Assim, SPs e IdPs teriam como definir quais tipos de protocolos (OpenId Connect padrão ou estendidos) eles suportam.

Adicionalmente, seria interessante avaliar meios para melhorar a usabilidade do processo de disseminação com uso de *tokens* que contenham o certificado digital do usuário, de modo que o usuário não precise inserir sua chave no navegador, assim removendo esse trabalho manual e propenso a erro.

Também será investigado a possibilidade de adição de semântica no protocolo utilizado, de modo a facilitar a adaptação dos sistemas e permitir a integração entre sistemas (IdP-SP) mais amigável e independente.

REFERÊNCIAS

ALLIANCE, C. Security guidance for critical areas of focus in cloud computing v3. 0. *Cloud Security Alliance*, 2011.

APACHE. *Apache CloudStack*. 2015. Acessado em 4 de fevereiro de 2015. <http://cloudstack.apache.org/>.

APACHE. *Overlays*. 2015. Acessado em 4 de fevereiro de 2015. <http://maven.apache.org/plugins/maven-war-plugin/overlays.html>.

BENANTAR, M. *Access control systems: security, identity management and trust models*. [S.l.]: Springer, 2006.

BERTINO, E.; TAKAHASHI, K. *Identity Management: Concepts, Technologies, and Systems*. [S.l.]: Artech House, 2011.

BETGÉ-BREZETZ, S. et al. End-to-end privacy policy enforcement in cloud infrastructure. In: IEEE. *Cloud Networking (CloudNet), 2013 IEEE 2nd International Conference on*. [S.l.], 2013. p. 25–32.

BETGÉ-BREZETZ, S. et al. Privacy control in the cloud based on multilevel policy enforcement. In: IEEE. *Cloud Networking (CLOUDNET), 2012 IEEE 1st International Conference on*. [S.l.], 2012. p. 167–169.

BIRRELL, E.; SCHNEIDER, F. B. Federated identity management systems: A privacy-based characterization. *IEEE security & privacy*, IEEE Computer Society, v. 11, n. 5, p. 36–48, 2013.

CAFE. *Cafe – Comunidade acadÃªmica federada*. 2007. Acessado em 3 de fevereiro de 2015. <http://portal.rnp.br/web/servicos/cafe>.

CANADÁ, M. da Justiça do. Personal information protection and electronic documents act. 2011. Acessado em 27 de fevereiro de 2014. <http://laws.justice.gc.ca/PDF/P-8.6.pdf>.

CHADWICK, D. W. Federated identity management. In: *Foundations of Security Analysis and Design V*. [S.l.]: Springer, 2009. p. 96–120.

CHADWICK, D. W.; FATEMA, K. A privacy preserving authorisation system for the cloud. *Journal of Computer and System Sciences*, Elsevier, v. 78, n. 5, p. 1359–1373, 2012.

CIVIL, C. Constituição da república federativa do brasil de 1988. Centro Gráfico do Senado Federal Brasília, 1988. Acessado em 28 de fevereiro de 2014. <http://www.planalto.gov.br/ccivil_03/constituicao/constituicao.htm>.

CIVIL, C. Lei nº12.965, de 23 abril de 2014. Centro Gráfico do Senado Federal Brasília, 2014. Acessado em 28 de fevereiro de 2014. <http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2014/lei/l12965.htm>.

CLERCQ, J. D. Single sign-on architectures. In: *Infrastructure Security*. [S.l.]: Springer, 2002. p. 40–58.

COMMISSION, U. S. F. T. Children's online privacy protection rule. 2013. Acessado em 27 de fevereiro de 2014. <http://www.gpo.gov/fdsys/pkg/FR-2013-12-20/html/2013-30293.htm>.

CONGRESS, U. S. Health insurance portability and accountability act of 1996. 1996. Acessado em 27 de fevereiro de 2014. <http://www.gpo.gov/fdsys/pkg/PLAW-104publ191/html/PLAW-104publ191.htm>.

CONGRESS, U. S. Gramm-leach-bliley act. 1999. Acessado em 27 de fevereiro de 2014. <http://www.gpo.gov/fdsys/pkg/PLAW-106publ102/html/PLAW-106publ102.htm>.

COUNCIL, H. R. The promotion, protection and enjoyment of human rights on the internet (a/hrc/20/l.13). 2012. Acessado em 09 de fevereiro de 2014. <http://ap.ohchr.org/documents/alldocs.aspx?doc_id=20280>.

DEBIAN. *Obtendo o Debian*. 2014. Acessado em 4 de fevereiro de 2015. <https://www.debian.org/distrib/>.

DENG, M. et al. A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. *Requirements Engineering*, Springer, v. 16, n. 1, p. 3–32, 2011.

DIAZ, C.; GÜRSES, S. Understanding the landscape of privacy technologies. *Extended abstract of invited talk in proceedings of the Information Security Summit*, p. 58–63, 2012.

DIRECTIVE, E. 95/46/ec of the european parliament and of the council of 24 october 1995 on the protection of individuals

with regard to the processing of personal data and on the free movement of such data. *Official Journal of the EC*, v. 23, n. 6, 1995. Acessado em 27 de fevereiro de 2014. <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML>.

ECLIPSE. *Higgins - Personal Data Service*. 2014. Acessado em 14 de Julho de 2014. <http://www.eclipse.org/higgins/>.

ECLIPSE. *Eclipse IDE for Java EE Developers*. 2015. Acessado em 4 de fevereiro de 2015. <https://eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/keplersr2>.

ENTR'OUVERT. *Authentic 2*. 2014. Acessado em 14 de Julho de 2014. <http://www.entrouvert.com/en/digital-identity/products>.

FORGEROCK. *Authentication and Authorization Overview*. 2011. Acessado em 15 de Julho de 2014. <https://wikis.forgerock.org/confluence/display/openam/Authentication+and+

FORGEROCK. *OpenAM*. 2014. Acessado em 14 de Julho de 2014. <http://forgerock.com/products/open-identity-stack/openam/>.

FORUM, C. I. *UK Cloud adoption and trends for 2013*. [S.l.], 2013. <http://cloudindustryforum.org/downloads/whitepapers/cif-white-paper-8-2012-uk-cloud-adoption-and-2013-trends.pdf>.

GOLDBERG, I. Privacy-enhancing technologies for the internet, ii: Five years later. In: SPRINGER. *Privacy Enhancing Technologies*. [S.l.], 2003. p. 1–12.

GOLDBERG, I.; WAGNER, D.; BREWER, E. *Privacy-enhancing technologies for the Internet*. [S.l.], 1997.

GOOGLE. *OpenID Connect (OAuth 2.0 for Login)*. 2014. Acessado em 3 de fevereiro de 2015. <https://developers.google.com/accounts/docs/OpenIDConnect>.

HALL, P. Opportunities for csps in enterprise-grade public cloud computing. *OVUM, May*, 2012.

HAN-ZHANG, W.; LIU-SHENG, H. An improved trusted cloud computing platform model based on daa and privacy ca scheme. In: *Computer Application and System Modeling (ICCASM), 2010 International Conference on*. [S.l.: s.n.], 2010. v. 13, p. V13–33–V13–39.

HANSEN, M. et al. Privacy-enhancing identity management. *Information Security Technical Report*, Elsevier, v. 9, n. 1, p. 35–44, 2004.

HANSEN, M.; SCHWARTZ, A.; COOPER, A. Privacy and identity management. *Security & Privacy, IEEE*, IEEE, v. 6, n. 2, p. 38–45, 2008.

HELFT, M. After breach, companies warn of e-mail fraud. *The New York Times*, Abril 2011. Acessado em 07 de fevereiro de 2014. <http://www.nytimes.com/2011/04/05/business/05hack.html>.

HERNAN, S. et al. Uncover security design flaws using the stride approach. *URL:http://msdn.microsoft.com/en-us/magazine/cc163519.aspx*, November 2006. Acessado em 07 de Janeiro de 2015.

HURSTI, J. Single sign-on. In: *Proc. Helsinki Uiniversity of Technology Seminar on Network Security*. [S.l.: s.n.], 1997.

IDENTITY, P. *PingFederate*. 2014. Acessado em 14 de Julho de 2014. <https://www.pingidentity.com/en/products/pingfederate.html>.

IETF. *The OAuth 2.0 Authorization Framework*. 2012. Acessado em 10 de Novembro de 2014. <http://tools.ietf.org/html/rfc6749#section-3.3>.

JANSEN, W.; GRANCE, T. et al. Guidelines on security and privacy in public cloud computing. *NIST special publication*, v. 800, p. 144, 2011.

JØSANG, A.; POPE, S. User centric identity management. In: CITESEER. *AusCERT Asia Pacific Information Technology Security Conference*. [S.l.], 2005. p. 77.

KALISKI, B. Pkcs# 5: Password-based cryptography specification version 2.0. 2000.

KOCIENIEWSKI, D. Adobe announces security breach. *The New York Times*, Outubro 2013. Acessado em 07 de fevereiro de 2014. <http://www.nytimes.com/2013/10/04/technology/adobe-announces-security-breach.html>.

LANDWEHR, C. et al. Privacy and cybersecurity: The next 100 years. *Proceedings of the IEEE*, IEEE, v. 100, n. Special Centennial Issue, p. 1659–1673, 2012.

LAUTERPACHT, H. Universal declaration of human rights, the. *Brit. YB Int'l L.*, HeinOnline, v. 25, p. 354, 1948.

LEE, H.; JEUN, I.; JUNG, H. Criteria for evaluating the privacy protection level of identity management services. In: IEEE. *Emerging Security Information, Systems and Technologies, 2009. SECURWARE'09. Third International Conference on.* [S.l.], 2009. p. 155–160.

LINDEN, M.; VILPOLA, I. An empirical study on the usability of logout in a single sign-on system. In: *Information Security Practice and Experience.* [S.l.]: Springer, 2005. p. 243–254.

MELL, P.; GRANCE, T. The nist definition of cloud computing. Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, 2011.

MIT. *MITREid Connect.* 2014. Acessado em 09 de Outubro de 2014. <https://github.com/mitreid-connect/OpenID-Connect-Java-Spring-Server>.

OECD, C. *Recommendation of the Council concerning Guidelines governing the Protection of Privacy and Transborder Flows of Personal Data (2013).* 2013. Acessado em 28 de junho de 2014. <http://www.oecd.org/sti/ieconomy/2013-oecd-privacy-guidelines.pdf>.

OPENID. *Libraries, Products, and Tools.* 2014. Acessado em 14 de Julho de 2014. <http://openid.net/developers/libraries/>.

OPENID. *OpenID Connect Core 1.0.* 2014. Acessado em 14 de Julho de 2014. <http://openid.net/specs/openid-connect-core-1_0.html>.

OPENID. *Welcome to OpenID Connect.* 2014. Acessado em 14 de Julho de 2014. <http://openid.net/connect/>.

ORAWIWATTANAKUL, T. et al. User-controlled privacy protection with attribute-filter mechanism for a federated sso environment using shibboleth. In: IEEE. *P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2010 International Conference on.* [S.l.], 2010. p. 243–249.

OSF, O. S. F. *Data Loss Statistics.* 2014. Acessado em 29 de junho de 2014. <http://datalossdb.org/statistics>.

PAREDES, L. N.; ZORZO, S. D. Privacy mechanism for applications in cloud computing. *Latin America Transactions, IEEE (Revista IEEE America Latina)*, IEEE, v. 10, n. 1, p. 1402–1407, 2012.

RENASIC. *Rede Nacional de SeguranÃ§a da InformaÃ§Ã£o e Criptografia*. 2014. Acessado em 10 de Novembro de 2014. <http://renasic.org.br/>.

SÁNCHEZ, R. et al. Enhancing privacy and dynamic federation in idm for consumer cloud computing. *Consumer Electronics, IEEE Transactions on*, IEEE, v. 58, n. 1, p. 95–103, 2012.

SANG-HUN, C. Theft of data fuels worries in south korea. *The New York Times*, Janeiro 2014. Acessado em 07 de fevereiro de 2014. <http://www.nytimes.com/2014/01/21/business/international/theft-of-data-fuels-worries-in-south-korea.html>.

SHIBBOLETH. *What's Shibboleth?* 2014. Acessado em 14 de Julho de 2014. <https://shibboleth.net/about/>.

SHOSTACK, A. *Threat Modeling: Designing for Security*. [S.l.]: John Wiley & Sons, 2014.

SMARI, W. W.; CLEMENTE, P.; LALANDE, J.-F. An extended attribute based access control model with trust and privacy: Application to a collaborative crisis management system. *Future Generation Computer Systems*, Elsevier, v. 31, p. 147–168, 2014.

SRINIVASAMURTHY, S.; LIU, D. Survey on cloud computing security. In: *Proc. Conf. on Cloud Computing, CloudCom*. [S.l.: s.n.], 2010. v. 10.

STARK, E.; HAMBURG, M.; BONEH, D. Symmetric cryptography in javascript. In: IEEE. *Computer Security Applications Conference, 2009. ACSAC'09. Annual*. [S.l.], 2009. p. 373–381.

STARK, E.; HAMBURG, M.; BONEH, D. *Stanford Javascript Crypto Library (SJCL)*. 2015. Acessado em 4 de fevereiro de 2015. <http://bitwiseshiftleft.github.io/sjcl/>.

STEVEN, J. Threat modeling-perhaps it's time. *Security & Privacy, IEEE*, IEEE, v. 8, n. 3, p. 83–86, 2010.

SWITCH. *About the Authentication and Authorization Infrastructure (AAI)*. 2004. Acessado em 3 de fevereiro de 2015. <https://www.switch.ch/aai/about/introduction/>.

SWITCH. *uApprove - User Consent Module for Shibboleth Identity Providers*. 2012. Acessado em 28 de junho de 2014. <https://www.switch.ch/aai/support/tools/uApprove.html>.

SWITCH. *Medium Demo*. 2014. Acessado em 14 de Julho de 2014. <https://www.switch.ch/aai/demo/2/medium.html>.

TERRELL, R. *An easy-to-use encryption system utilizing RSA and AES for javascript*. 2012. Acessado em 24 de maio de 2014. <https://github.com/wwwtyro/cryptico>.

WEINGÄRTNER, R.; WESTPHALL, C. M. Enhancing privacy on identity providers. *The Eighth International Conference on Emerging Security Information, Systems and Technologies – SECURWARE*, ThinkMind, 2014.

WU, T. *RSA and ECC in JavaScript*. 2009. Acessado em 14 de maio de 2014. <http://www-cs-students.stanford.edu/ tjw/jsbn/>.

XENPROJECT. *The Hypervisor*. 2015. Acessado em 4 de fevereiro de 2015. <http://www.xenproject.org/developers/teams/hypervisor.html>.

XENPROJECT. *XAPI*. 2015. Acessado em 4 de fevereiro de 2015. <http://www.xenproject.org/developers/teams/xapi.html>.

XIAO, Z.; XIAO, Y. Security and privacy in cloud computing. *Communications Surveys & Tutorials, IEEE*, IEEE, v. 15, n. 2, p. 843–859, 2013.

ZHANG, Q. et al. A study on context-aware privacy protection for personal information. In: IEEE. *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on*. [S.l.], 2007. p. 1351–1358.